Exploring ironmaking practice at Meroe, Sudan—images, data, and analytical scripts

Michael F. Charlton[1]
Jane Humphris[1]

[1]*UCL Qatar*

27 July 2017

Introduction

Meroe, Sudan was an important ironmaking centre for the Kingdom of Kush between the 7th Century BC and the 4th Century AD (Humphris and Rehren, 2014). Following excavation of a furnace workshop beneath the 2nd - 4th Century AD slag heap labeled MIS6, a series of iron production experiments were conducted in January 2015 in order to test technological models and investigate resource utilization. Analysis of smelting residues from the MIS6 workshop and the experiments subsequently took place at UCL Qatar in late 2016.

Methods

Residues were submitted to the UCL Qatar Archaeological Materials Laboratories in Doha, where they were catalogued, sectioned, cleaned, and mounted in epoxy resin blocks. Each specimen was polished following standard metallographic techniques to a finish of 1 µm. Specimens were investigated via optical microscopy and given a secondary material identification. These were then coated with carbon and re-examined with scanning electron microscopy (SEM) and energy dispersive (EDS) X-ray microanalysis.

Optical microscopy

Polished specimens were investigated with reflected light microscopy using a Leica DM 2500P at variable magnifications. Notes were taken and a series of images generated for later reference. Most of the images are not created at typical publication standards, but are provided in this dataset for their information content. Files consist of up to 6 stacked images in a TIFF format to enable quick examination of specimen variability. ***ImageJ*** (Schindelin et al., 2015) is the recommended software package for exploring the stacks.

Scanning Electron Microscopy and X-ray microanalysis

Carbon-coated specimens were explored with a JEOL JSM 6610 low vacuum SEM equipped with an Oxford Instruments X-Max N50 energy dispersive spectrometer (EDS). Images were captured with backscattered electrons to observe phase differences, and element compositional analyses conducted for each. X-ray spectra were acquired, interrogated, and quantified using Oxford Instruments Aztec 3.1. Spectra were optimized using a cobalt standard and acquisition parameters were kept constant (working distance = 10 mm; accelerating voltage = 20kV; process time = 5; deadtime ≈ 40%). All EDS analyses were conducted at magnifications of x200 or greater, most comprising area scans at x200. Peak identifications were made manually in order to optimize each fitted spectrum to its empirical spectrum and identify anomalies created by the pulse pile up correction algorithm. Images for all area analyses are provided in this dataset and can be compared with the raw analytical data for each in the *.RData*, *.txt*, and *.xlsx* files named ***Meroe XP and MIS6 SEM-EDS data***. Files consist of 5 analytical images stacked into a TIFF format to enable quick examination of specimen variability. ***ImageJ*** (Schindelin et al., 2015) is the recommended software package for exploring the stacks.

Data quality

Three certified reference materials (CRMs) were analysed during most analytical sessions to ensure good instrument performance and provide a means of assessing the precision and accuracy of quantified results. The US Geological Survey CRM basalts BCR-2G (Plumlee, 1998a), BIR-1G (Smith, 1998), and BHVO-2G (Plumlee, 1998b), were selected. Results deviated from the recommended values and between runs due to variations in beam current and minor differences in focusing. Normalization across all identified elements, however, leads to general increases in accuracies and precision for most elements with recommended values greater than 0.1 wt %. Interested readers are invited to explore the data acquired for each CRM during this study as provided in the *.RData*, *.txt*, and *.xlsx* files named ***CRM Analyses.***

Data analysis

A key component of the Sudan Project's activities at Meroe is to identify the resources exploited by ancient smelters. This task can be complicated by the fact that iron production generates a range of material products with blurred definitions. Following macro- and microscopic materials characterization, data must be sorted into chemical groups that enable of comparisons between functionally similar materials.

The script and output below provide a detailed account of how the experimental and archaeological ironmaking residues were sorted into chemical groups for the aims of this project. The approach taken relies on the iterative use of principal component analysis (PCA) and hierarchical cluster analysis (HCA). A final set of comparisons are made with data converted into oxides via stoichiometry.

Code for the specialized plotting and analysis functions can be found in the attached script and PDF documents named ***Custom plotting and related functions for compositional data analysis***. All analyses were written in RStudio 1.0.146 and performed using R version 3.4.1 (R Core Team, 2017). In addition to the base package, scripts require functions found in the MASS (Venables and Ripley, 2002) and robCompositions (Templ et al., 2011) libraries. Users are invited to explore our results using the code below by applying it to the associated dataset. They are also encouraged to modify related functions and scripts to explore this and other datasets with similar data. However, caution is warranted when comparing these analyses to those produced by of other laboratories and projects without due consideration to data quality biases.

Load relevant packages:

```
library(MASS)   ## Multivariate analysis functions
library(robCompositions)  ## functions for log-ratio compositional analysis
```

Clear variables and data from previous R sessions

```
rm(list=ls(all=TRUE))
```

Load custom functions

Either run the functions from appendix 1 or load a workspace containing the functions. The latter is recommended.

```
load(file=file.choose())
```

Select variables of interest

```
data.meta<-c(
  "lab.spec",
  "specimen",
  "image",
  "context",
  "material",
  "type",
  "Date",
  "measure",
  "unit"
)

els.all<-c(
  "O",
```

```
  "Na",
  "Mg",
  "Al",
  "Si",
  "P",
  "S",
  "Cl",
  "K",
  "Ca",
  "Sc",
  "Ti",
  "V",
  "Cr",
  "Mn",
  "Fe",
  "Co",
  "Ni",
  "Cu",
  "Zn",
  "As",
  "Sr",
  "Zr",
  "Mo",
  "Ba",
  "Ce",
  "Nd",
  "Os"
)

els.ox<-c(
  "Mg",
  "Al",
  "Si",
  "P",
  "K",
  "Ca",
  "Ti",
  "Mn",
  "Fe",
  "Zr",
  "Na",
  "Ba",
```

```
  "Co"
)


ox.per<-c(
  "MgO",
  "Al2O3",
  "SiO2",
  "P2O5",
  "K2O",
  "CaO",
  "TiO2",
  "MnO",
  "FeO",
  "Zr",
  "Na2O",
  "BaO",
  "Co"
)

tern.els<-c(
  "FeO",
  "MnO",
  "SiO2",
  "Al2O3"
)
tern.poles<-c(
  "FeO.MnO",
  "Al2O3",
  "SiO2"
)
```

Load data

```
data<-read.csv(file=file.choose(),header=TRUE, row.names=1)
```

Normalize data

Short term drift in the electron beam current of scanning electron microscopes introduces variation in quantitative measures of element composition. The normalization process assumes that such measurement noise is uniform across all beam measures with totals approaching 100 wt%. Analysis of fused basalt CRMs shows this to be the case. Additional error is introduced by

imperfect sample preparation, the presence of voids, and sample matrix. The degree to which normalization also corrects for these is unknown, though similarly prepared samples with equivalent matrices are expected to exhibit comparable levels of uncertainty.

```
data.meas<-data[data$measure=="measure" | data$measure=="single",]
sum.data<-apply(data.meas[els.all],1,sum)
norm.data<-data.meas[els.all]/sum.data*100
data.meas[els.all]<-norm.data
```

Calculate sample means

In general, chemical composition was measured in five areas for each specimen. Mean chemistries are assumed to be characteristic of the specimen as a whole and are calculated prior to making further comparisons.

```
spec<-NULL
spec.mean<-NULL
for(i in 1:nlevels(data.meas$lab.spec)){
  spec[[i]]<-data.meas[data.meas$lab.spec==levels(data.meas$lab.spec)[i],]
  for(j in 1:length(spec)){
    spec.mean[[j]]<-colMeans(spec[[j]][els.all])
  }
}

data.mean<-data.meas[FALSE,]
for(m in 1:length(spec.mean)){
  data.mean[m,els.all]<-t(as.data.frame(spec.mean[[m]]))
  data.mean[m,data.meta]<-as.data.frame(spec[[m]][nrow(spec[[m]]),data.meta])
  data.mean$measure[m]<-"mean"
  data.mean$image[m]<-"mean"
}

rownames(data.mean)<-data.mean$lab.spec
```

Calculate sample standard deviations for each specimen

The calculation of specimen standard deviations facilitates the assessment of variable uncertainty.

```
data.sdev<-data.mean[FALSE,]
for(k in 1:length(spec)){
  for(l in 1:length(els.all)){
    data.sdev[k,els.all][,l]<-sd(spec[[k]][els.all][,l])
```

```
    data.sdev[k,data.meta]<-data.mean[k,data.meta]
    data.sdev$measure[k]<-"sdev"
    data.sdev$image[k]<-"sdev"
  }
}

rownames(data.sdev)<-paste(rownames(data.mean),"_sdev",sep="")
```

Impute rounded zeros

Missing variables arising from each area scan were assumed to be below detection. These were given a nominal value of zero. Where the variable mean is exactly equal to zero across all, that variable is not considered for further analysis. Otherwise, all zeros are imputed using the **impRZilr()** function from the robCompositions package(Templ et al., 2011). The elements O and Cl are then removed from interest because it is assumed that they add noise to the residues of metallurgical systems. The rest become part of the working data in the dataframe ***data.wk***.

```
els.imp<-c()
n<-0
for(t in 1:length(els.all)){
  if(colMeans(data.mean[els.all])[t]==0){
    next
  }else{
    n<-n+1
    els.imp[n]<-els.all[t]
  }
}


data.imp<-impRZilr(data.mean[els.imp],dl=rep(0.001,ncol(data.mean[els.imp])))
data.mean[els.imp]<-data.imp$x

els<-els.imp[which(els.imp!=c("O","Cl"))]

data.wk<-data.mean
```

Multivariate analysis of ore chemistry

Identification of furnace workshop ores

Potential ore residues were sampled from MIS6-FW224, the furnace fill. All of these were assumed to be partially reacted. In order to determine which were closer to the actual ore used, a systematic examination and sorting of materials from the furnace fill was undertaken.

select relevant data

```
fw<-data.wk[data.wk$context=="MIS6-FW224",]
```

select elements of interest

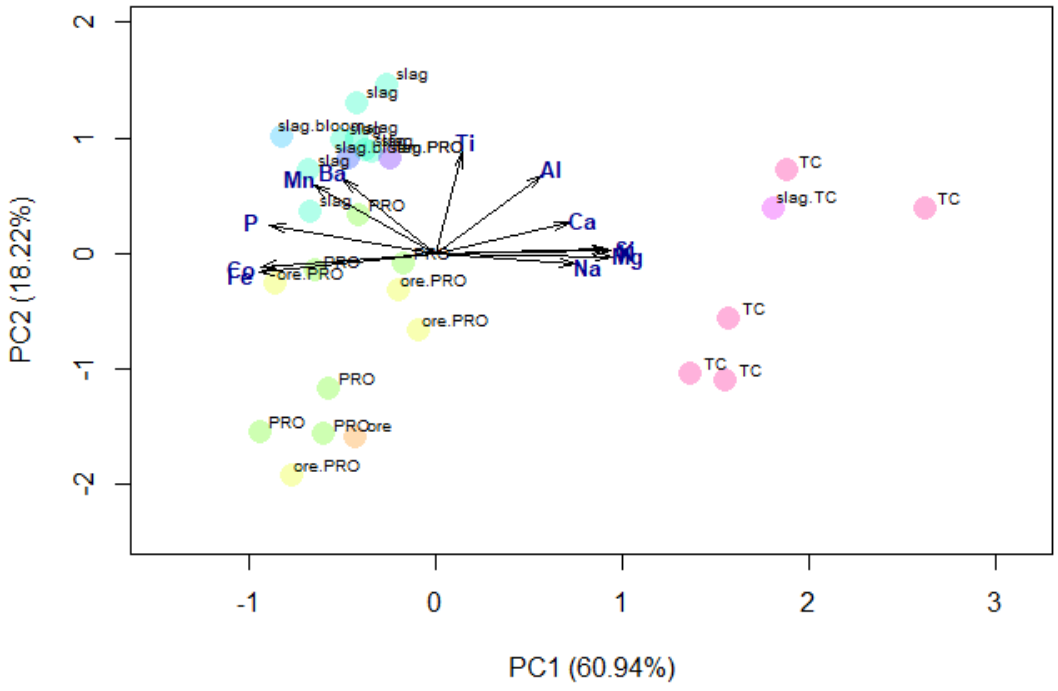
```
els.fw<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(fw[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    els.fw[n]<-els[t]
  }
}
```

transform data to subcompositions

```
sum.fw<-apply(fw[els.fw],1,sum)
fw[els.fw]<-fw[els.fw]/sum.fw*100
```

plot PCA scores and loadings

```
pcaplot(fw[els.fw],1,2,fw$material,fw$material)
```



remove technical ceramics from consideration

```
fw.orsl<-data.wk[data.wk$context=="MIS6-FW224" &
                   data.wk$material!="TC" &
                   data.wk$material!="slag.TC",]
```

select elements of interest for material subset

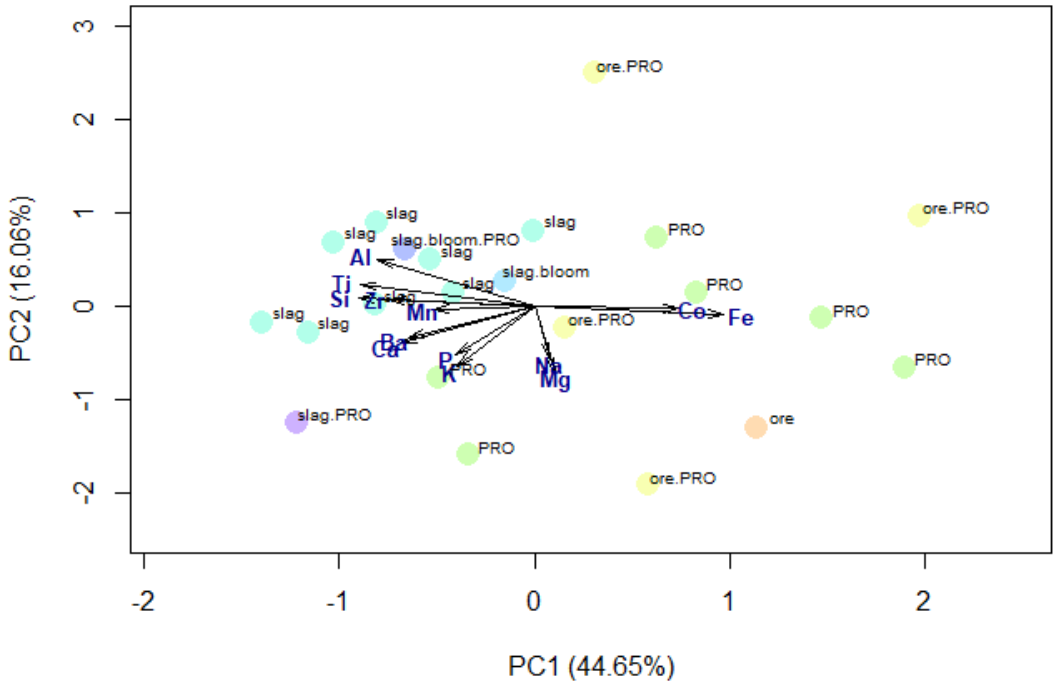
```
els.orsl<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(fw.orsl[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    els.orsl[n]<-els[t]
  }
}
```

transform data to subcompositions

```
sum.orsl<-apply(fw.orsl[els.orsl],1,sum)
fw.orsl[els.orsl]<-fw.orsl[els.orsl]/sum.orsl*100
```

plot PCA scores and loadings

```
pcaplot(fw.orsl[els.orsl],1,2,fw.orsl$material,fw.orsl$material)
```



remove slag from consideration

```
fw.pro<-data.wk[data.wk$context=="MIS6-FW224" &
                  (data.wk$material=="PRO" |
                     data.wk$material=="ore.PRO" |
                     data.wk$material=="ore"),]
```

select elements of interest for material subset

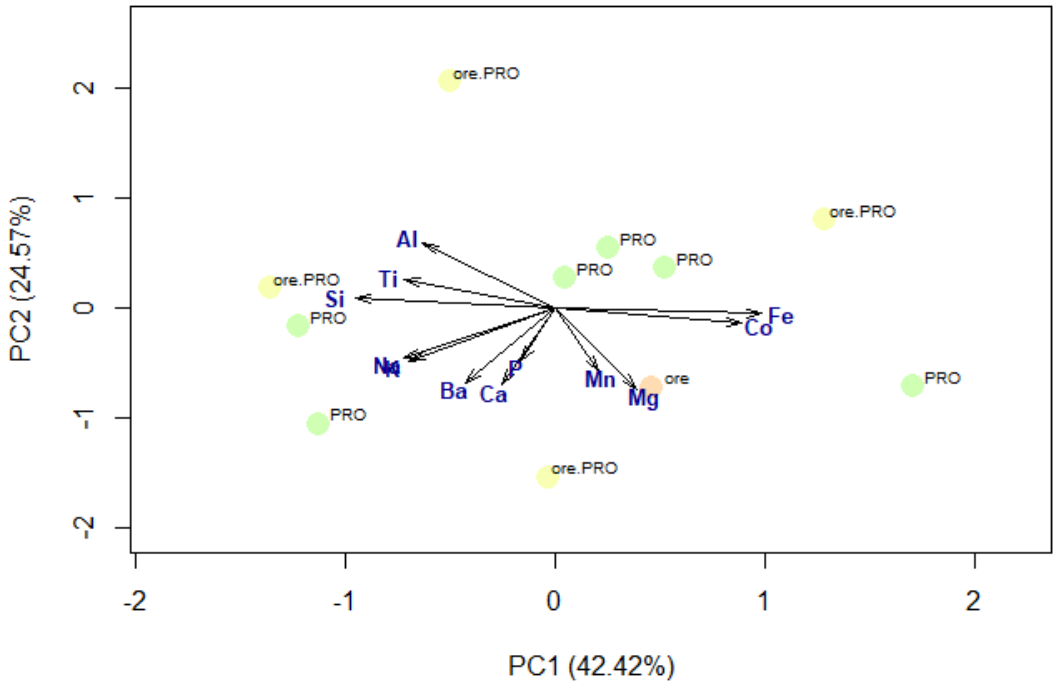
```
els.pro<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(fw.pro[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    els.pro[n]<-els[t]
  }
}
```

transform data to subcompositions

```
sum.pro<-apply(fw.pro[els.pro],1,sum)
fw.pro[els.pro]<-fw.pro[els.pro]/sum.pro*100
```

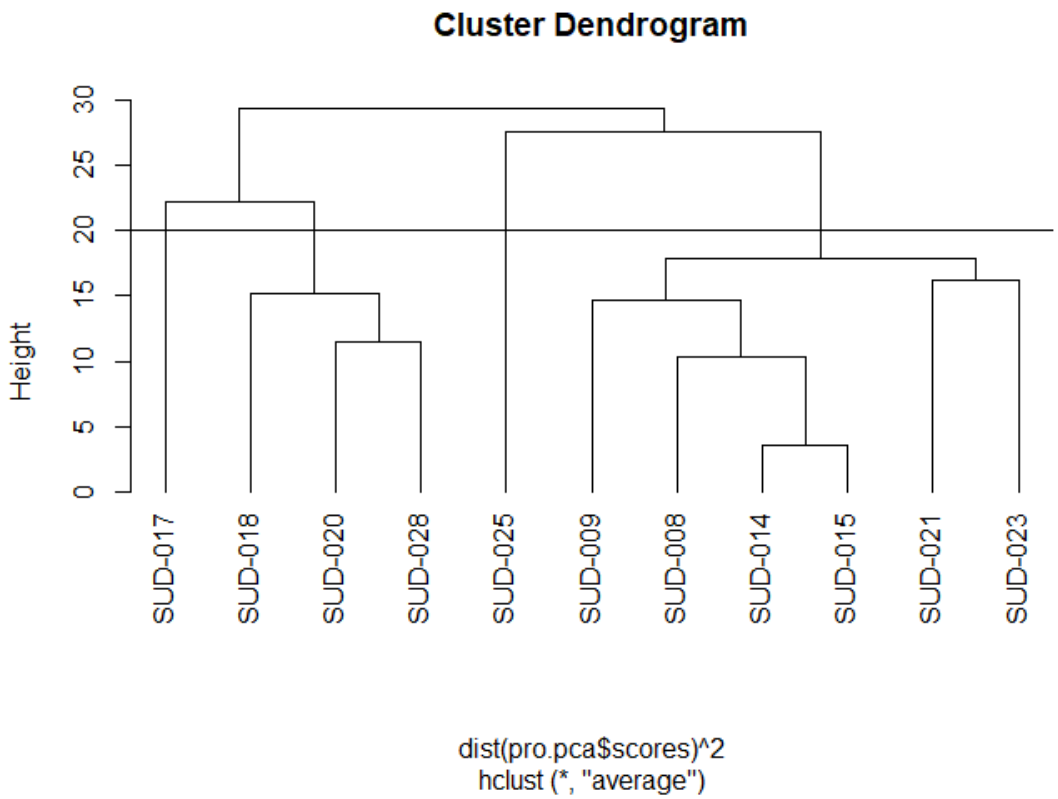
plot PCA scores and loadings

```
pro.pca<-pcaplot(fw.pro[els.pro],1,2,fw.pro$material,fw.pro$material)
```



apply average-linkage cluster analysis

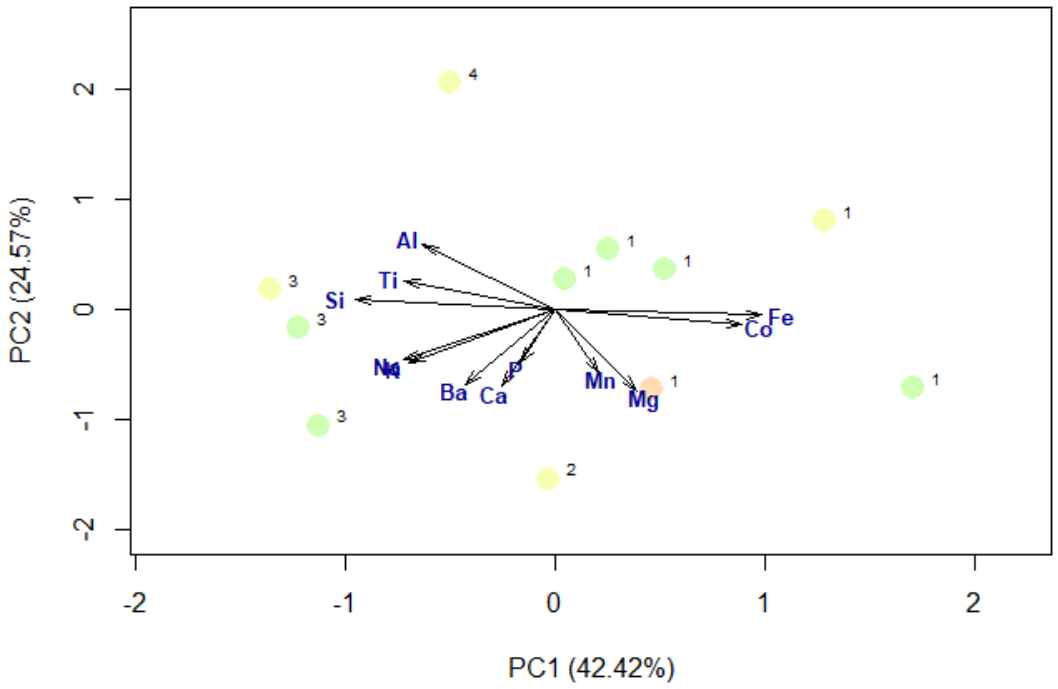
```
pro.hca<-hclust(dist(pro.pca$scores)^2,"ave")
plot(pro.hca,hang=-1)
abline(h=20)
```




```
pro.cluster<-cutree(pro.hca,h=20)
```

plot PCA scores labeled by cluster

```
pcaplot(fw.pro[els.pro],1,2,fw.pro$material,pro.cluster)
```



rename clusters and define chemical groups

```
fw.ore<-fw.pro[pro.cluster==1,]
pro<-fw.pro[pro.cluster!=1,]

fw.ore$chem.group<-rep("ore",nrow(fw.ore))
data.wk[rownames(fw.ore),
        "chem.group"]<-"ore"
pro$chem.group<-rep("PRO",nrow(pro))
data.wk[rownames(pro),"chem.group"]<-"PRO"
```

Compare experimental and archaeological ore chemistry

select ore data

```
ore.wk<-data.wk[data.wk$material=="ore" |
                  data.wk$chem.group=="ore" &
                  is.na(data.wk$chem.group)==F,]
```

select elements of interest for ore

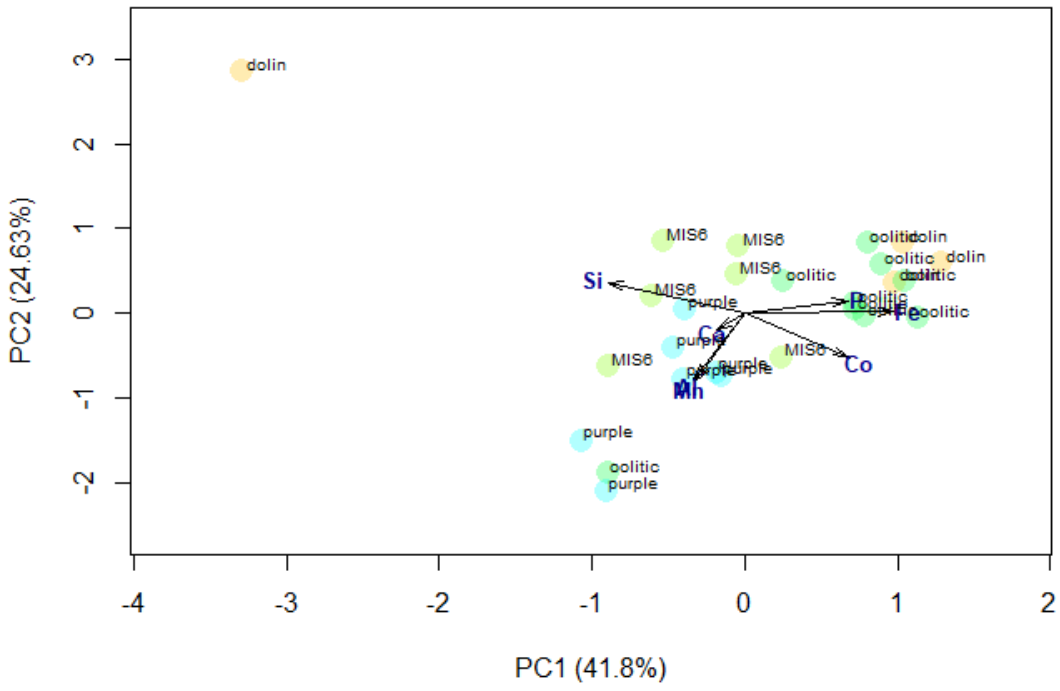
```
ore.els<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(ore.wk[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    ore.els[n]<-els[t]
  }
}
```

transform data to subcompositions

```
sum.ore.wk<-apply(ore.wk[ore.els],1,sum)
ore.wk[ore.els]<-ore.wk[ore.els]/sum.ore.wk*100
```

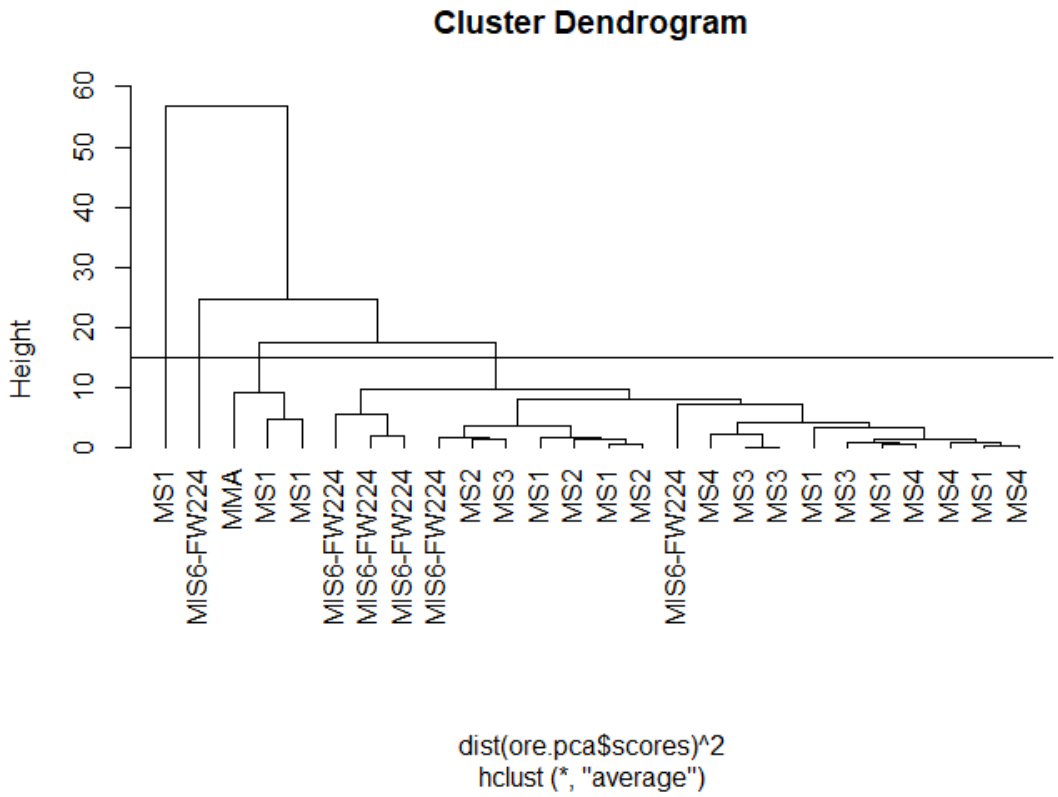
plot PCA scores and loadings

```
ore.pca<-pcaplot(ore.wk[ore.els],1,2,ore.wk$type,ore.wk$type)
```



apply average-linkage cluster analysis

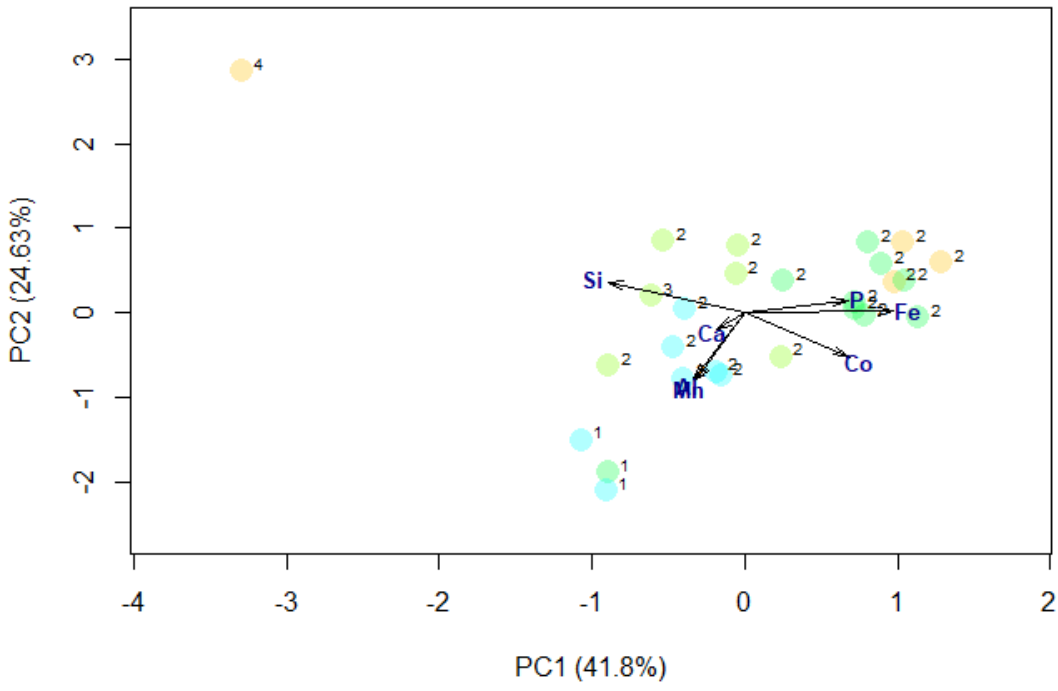
```
ore.hca<-hclust(dist(ore.pca$scores)^2,"ave")
plot(ore.hca,hang=-1,labels=ore.wk$context)
abline(h=15)
```




```
ore.cluster1<-cutree(ore.hca,h=15)
```

plot PCA scores labeled by cluster

```
ore.pca<-pcaplot(ore.wk[ore.els],1,2,ore.wk$type,ore.cluster1)
```



rename clusters and define chemical groups

```
data.wk[names(ore.cluster1[ore.cluster1==4]),]$chem.group<-"Dolin outlier"
data.wk[names(ore.cluster1[ore.cluster1==1]),]$chem.group<-"Hi-Al outlier"
data.wk[names(ore.cluster1[ore.cluster1==2]),]$chem.group<-"ore"
data.wk[names(ore.cluster1[ore.cluster1==3]),]$chem.group<-"Hi-Ca outlier"
```

remove outliers from consideration

```
ore.wk2<-data.wk[data.wk$chem.group=="ore" &
                   is.na(data.wk$chem.group)==F,]
ore.wk2<-droplevels(ore.wk2)
```

select elements of interest for subset

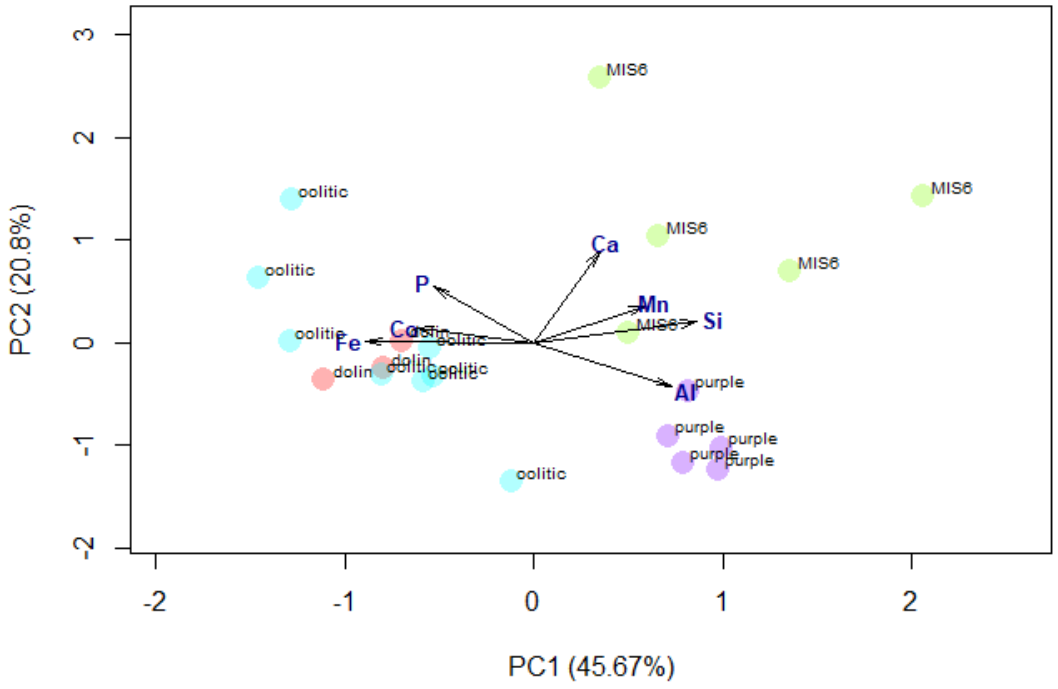
```
ore.els2<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(ore.wk2[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    ore.els2[n]<-els[t]
  }
}
```

transform data to subcompositions

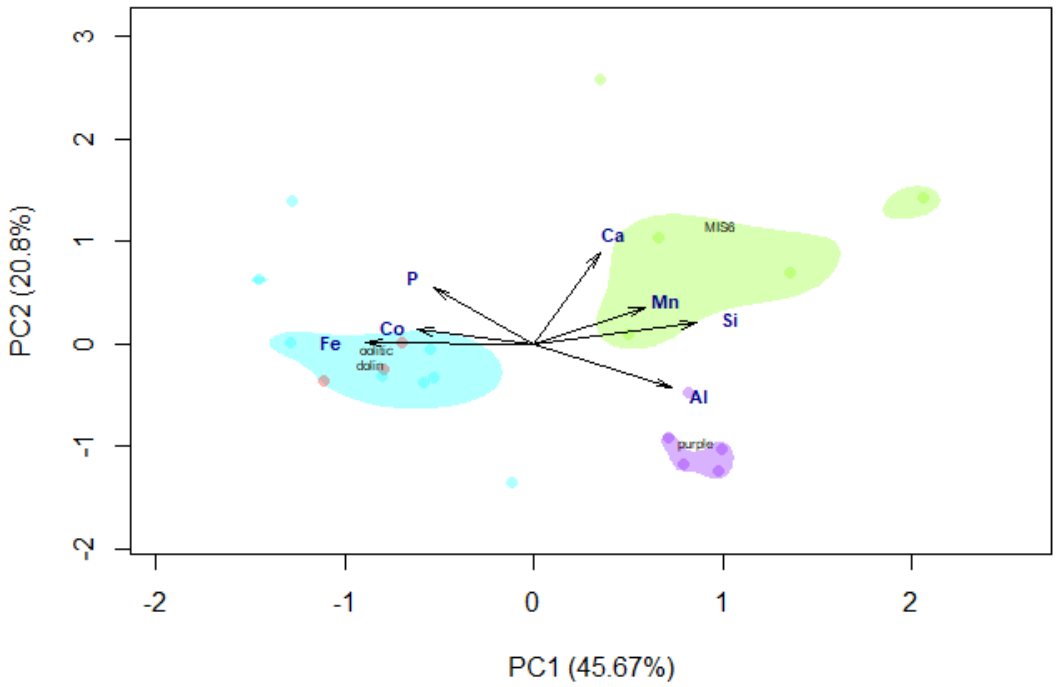
```
sum.ore.wk2<-apply(ore.wk[ore.els2],1,sum)
ore.wk2[ore.els2]<-ore.wk2[ore.els2]/sum.ore.wk2*100
```

plot PCA scores and loadings

```
ore.pca2<-pcaplot(ore.wk2[ore.els2],1,2,ore.wk2$type,ore.wk2$type)
```




```
pcaplot.kde(ore.wk2[ore.els2],1,2,ore.wk2$type,ore.pca2$fc)
```



reset ore data

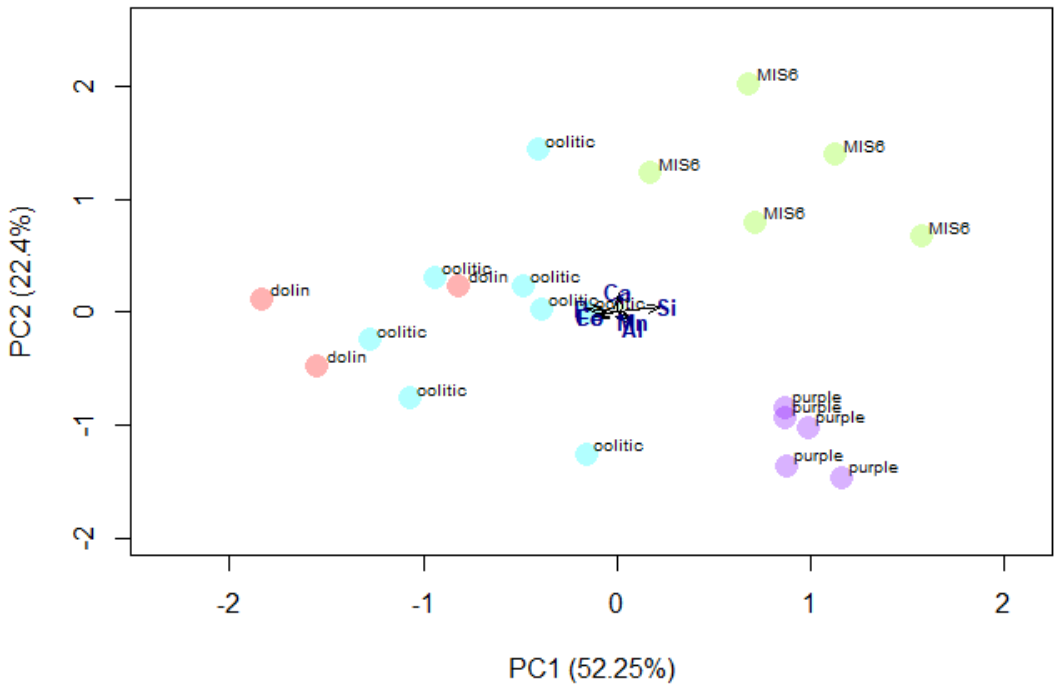
```
ore.wk2<-data.wk[data.wk$chem.group=="ore" &
                   is.na(data.wk$chem.group)==F,]
ore.wk2<-droplevels(ore.wk2)
```

transform data to centered log-ratios (CLR)

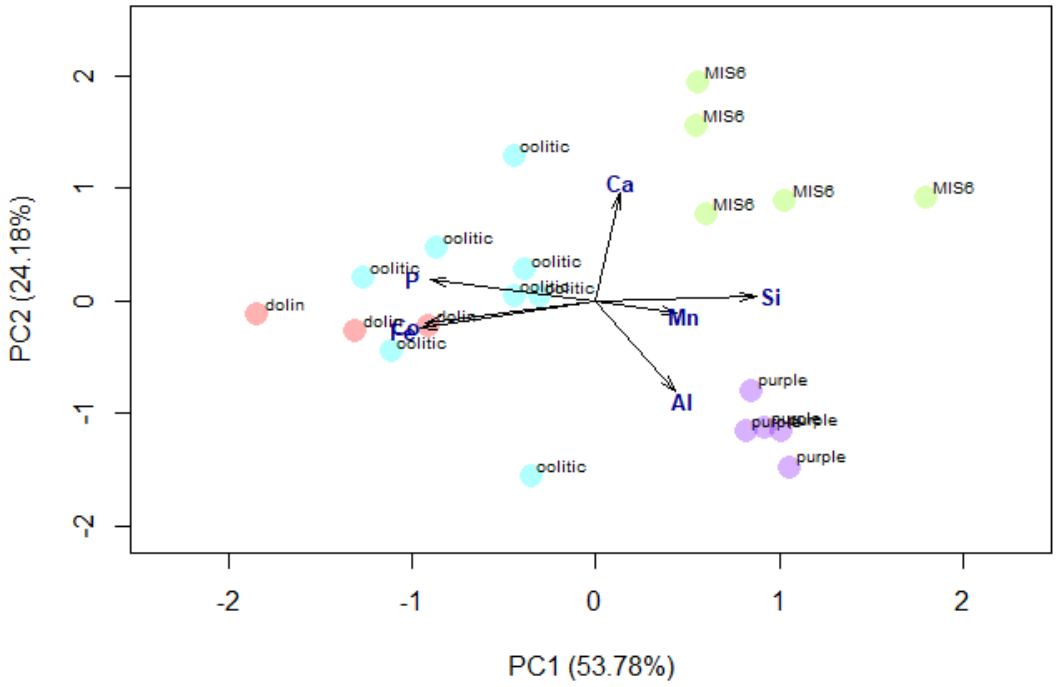
```
ore.gm<-apply(ore.wk2[ore.els2],1,geomean)
ore.clr<-log10(ore.wk2[ore.els2]/ore.gm)
```

create CLr and standardized CLR-PCA plots

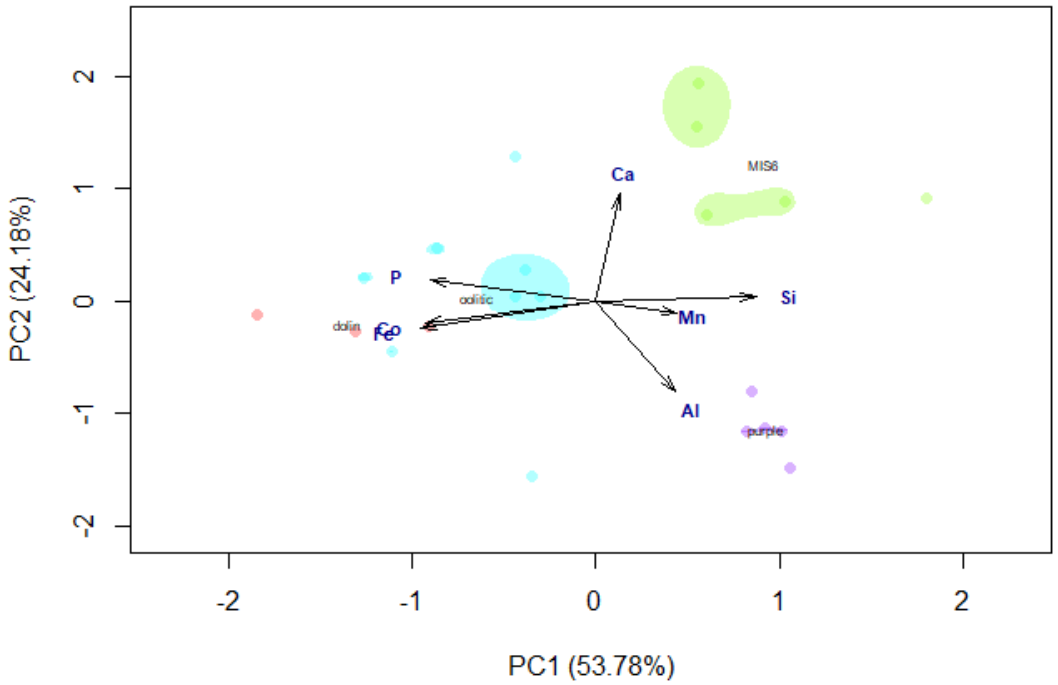
```
clr.ore<-pcaplot.clr(ore.clr,1,2,ore.wk2$type,ore.wk2$type)
```



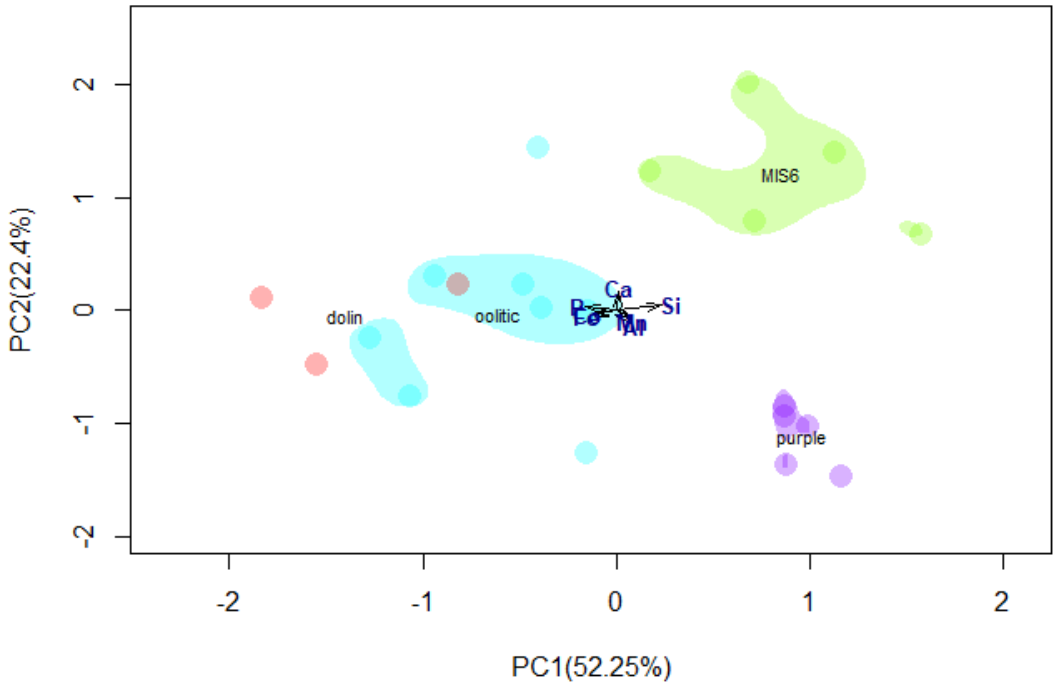
```
clr.ore2<-pcaplot(ore.clr,1,2,ore.wk2$type,ore.wk2$type)
```




```
pcaplot.kde(ore.clr,1,2,ore.wk2$type,clr.ore$fc)
```



```
pcaplot.clr.kde(ore.clr,1,2,ore.wk2$type,clr.ore$fc)
```



create CSV file of ore dataset

```
write.csv(ore.wk2,file=file.choose())
```

Multivariate analysis of slag

Identification of furnace workshop slag

Iron production residues were sampled from two contexts within the MIS6 furnace workshop: a) MIS6-FW224--furnace fill and b) MIS6-FW262--probable smithing hearth. These two types of slag, as well as other potential outliers, needed to be identified and separated prior to comparisons of smelting slag.

select workshop slag data

```
slag.wk<-data.wk[(data.wk$material=="slag" |
                  data.wk$material=="slag.PRO" |
                  data.wk$material=="slag.bloom" |
                  data.wk$material=="slag.bloom.PRO" |
                  data.wk$material=="slag.TC") &
                  data.wk$context!="MS5",]

slag.fw<-slag.wk[slag.wk$context=="MIS6-FW224" |
                  slag.wk$context=="MIS6-FW262",]
```

select elements of interest

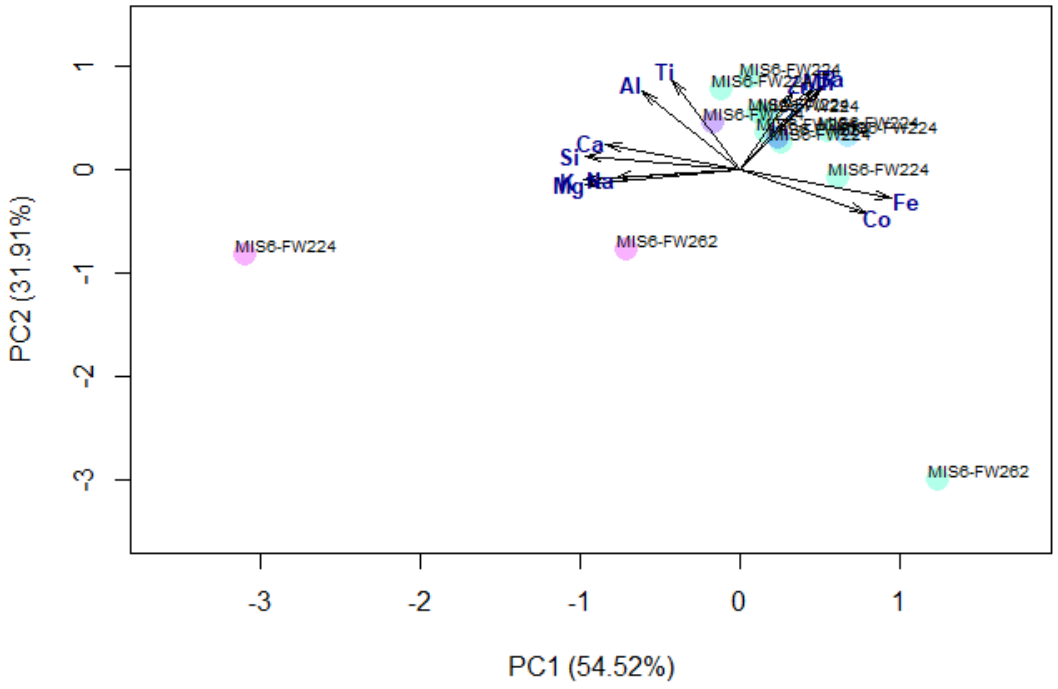
```
els.sfw<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(slag.fw[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    els.sfw[n]<-els[t]
  }
}
```

transform data to subcompositions

```
sum.slag.fw<-apply(slag.fw[els.sfw],1,sum)
slag.fw[els.sfw]<-slag.fw[els.sfw]/sum.slag.fw*100
```

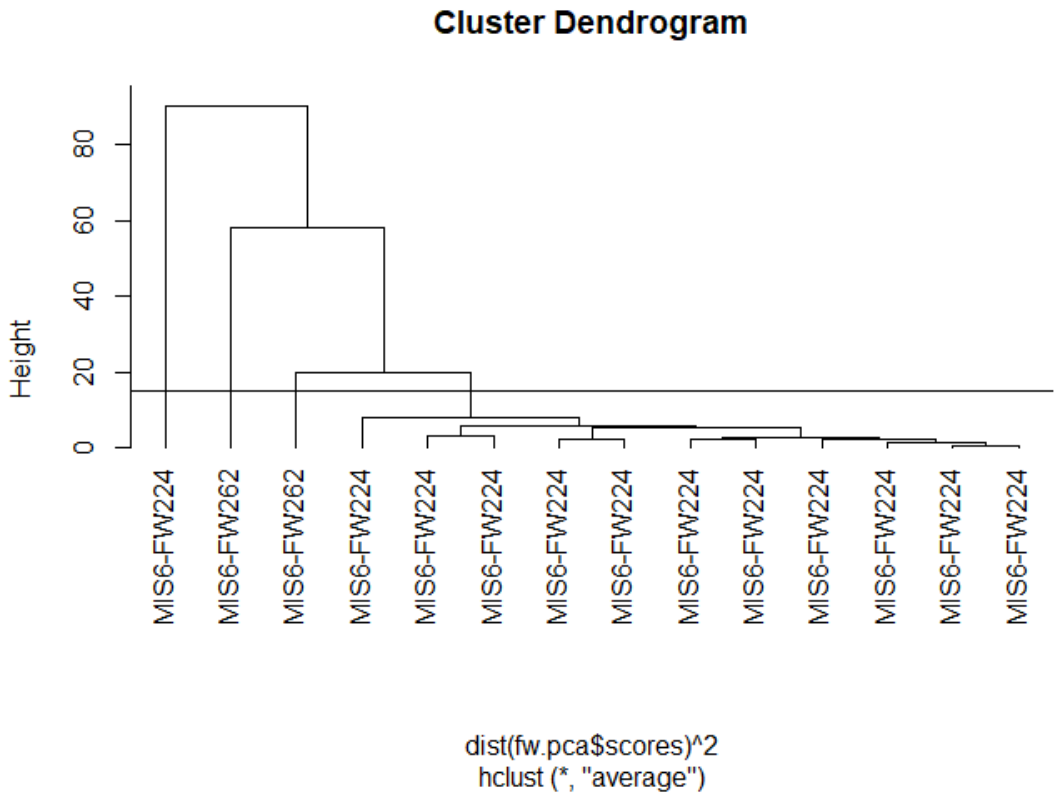
plot PCA scores and loadings

```
fw.pca<-pcaplot(slag.fw[els.sfw],1,2,slag.fw$material,slag.fw$context)
```



apply average-linkage cluster analysis

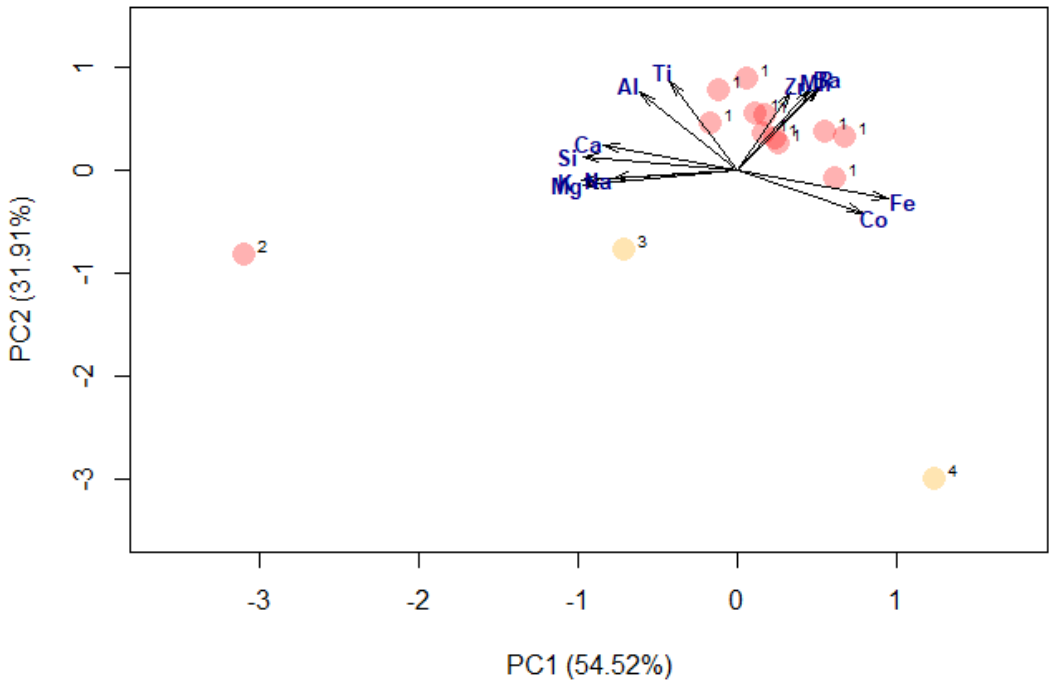
```
fw.slag.hc<-hclust(dist(fw.pca$scores)^2,"ave")
plot(fw.slag.hc,hang=-1,labels=slag.fw$context)
abline(h=15)
```




```
fw.slag.clust<-cutree(fw.slag.hc,h=15)
```

plot PCA scores labeled by cluster

```
pcaplot(slag.fw[els.sfw],1,2,slag.fw$context,fw.slag.clust)
```



rename clusters and define chemical groups

```
data.wk[names(fw.slag.clust[fw.slag.clust==2]),]$chem.group<-"FA slag"
data.wk[names(fw.slag.clust[fw.slag.clust==3 |
                              fw.slag.clust==4]),]$chem.group<-"smithing
slag"
data.wk[names(fw.slag.clust[fw.slag.clust==1]),]$chem.group<-"smelting slag"
```

Compare experimental and archaeological slag chemistry

select all slag data

```
slag.wk2<-data.wk[data.wk$material=="slag" &
                    (data.wk$chem.group=="smelting slag" |
                      is.na(data.wk$chem.group)==T) &
                    data.wk$context!="MS5",]
slag.wk2<-droplevels(slag.wk2)
```

select elements of interest

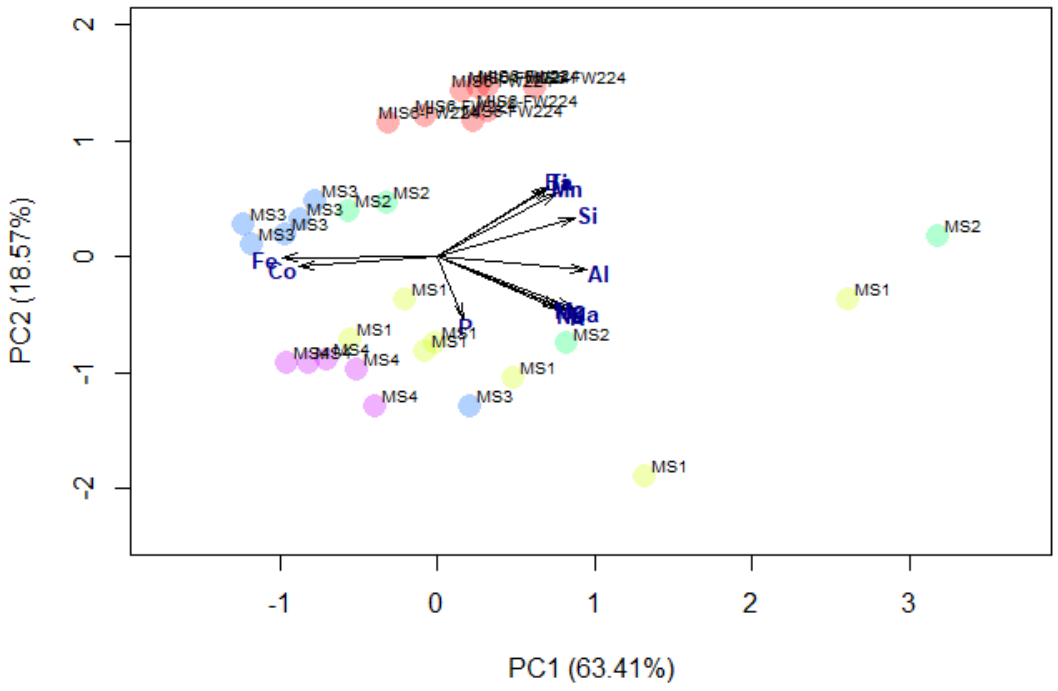
```
els.slag<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(slag.wk2[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    els.slag[n]<-els[t]
  }
}
```

transform data to subcompositions

```
sum.slag.wk2<-apply(slag.wk2[els.slag],1,sum)
slag.wk2[els.slag]<-slag.wk2[els.slag]/sum.slag.wk2*100
```

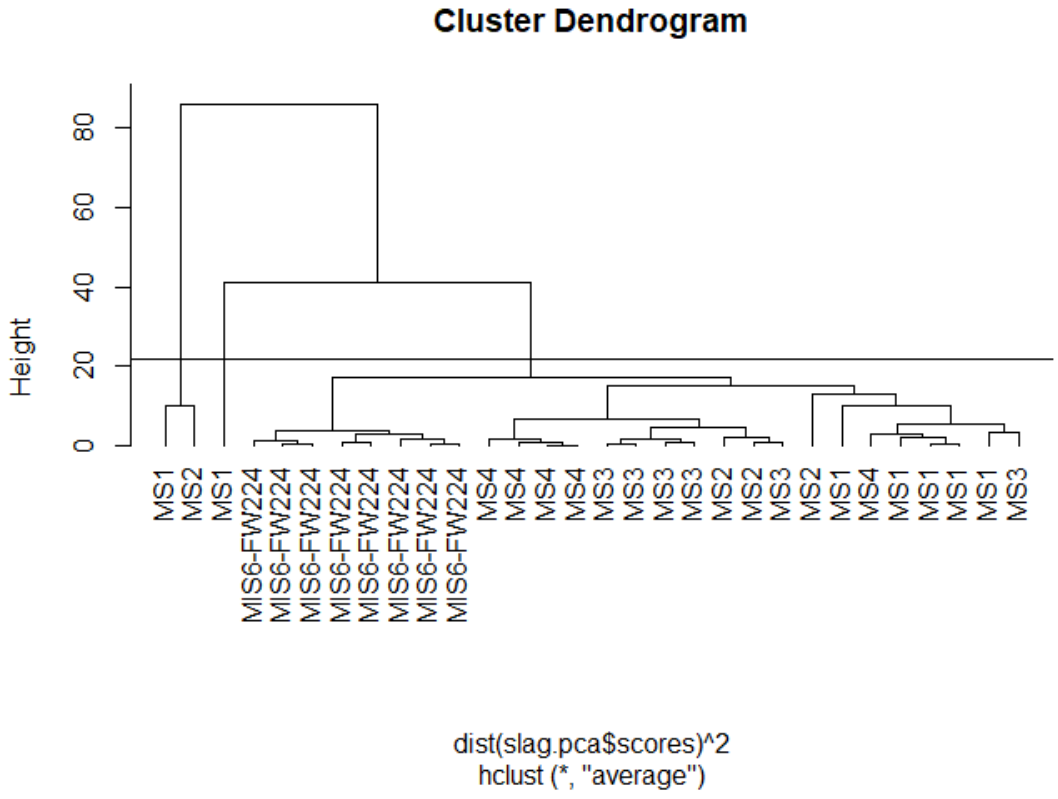
plot PCA scores and loadings

```
slag.pca<-pcaplot(slag.wk2[els.slag],1,2,slag.wk2$context,slag.wk2$context)
```



apply average-linkage cluster analysis

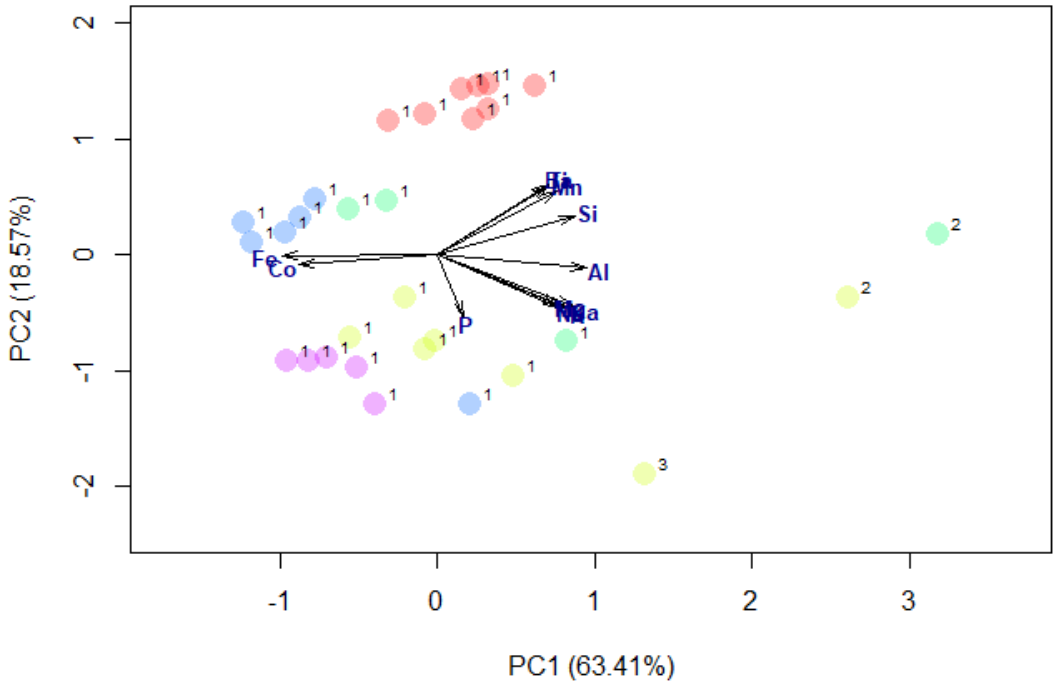
```
slag.hcl<-hclust(dist(slag.pca$scores)^2,"ave")
plot(slag.hcl,hang=-1,labels=slag.wk2$context)
abline(h=22)
```



```
slag.clust<-cutree(slag.hcl,h=22)
```

plot PCA scores by cluster

```
pcaplot(slag.wk2[els.slag],1,2,slag.wk2$context,slag.clust)
```



rename clusters and define chemical groups

```
data.wk[names(slag.clust[slag.clust==2]),]$chem.group<-"bloom slag"
data.wk[names(slag.clust[slag.clust==3]),]$chem.group<-"FA slag2"
```

remove non-smelting slag from consideration

```
slag.wk3<-data.wk[names(slag.clust)[slag.clust==1],]
slag.wk3<-droplevels(slag.wk3)
```

select elements of interest for subset

```
els.slag2<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(slag.wk3[els]))[t]<0.1){
    next
```

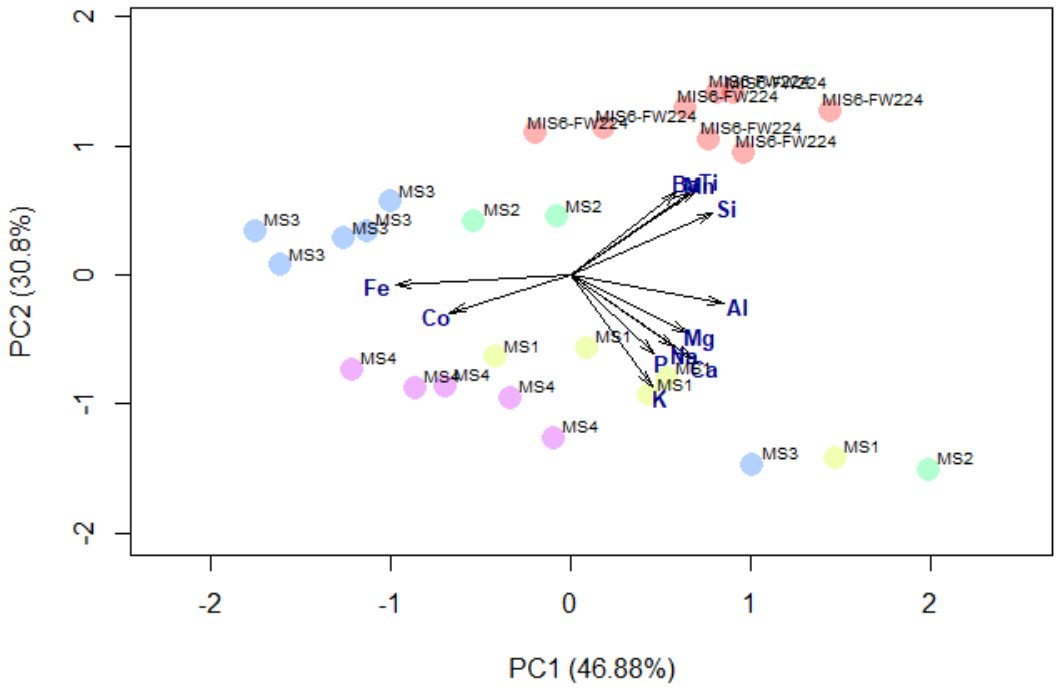
```
  }else{
    n<-n+1
    els.slag2[n]<-els[t]
  }
}
```

transform data to subcompositions

```
sum.slag.wk3<-apply(slag.wk3[els.slag2],1,sum)
slag.wk3[els.slag2]<-slag.wk3[els.slag2]/sum.slag.wk3*100
```

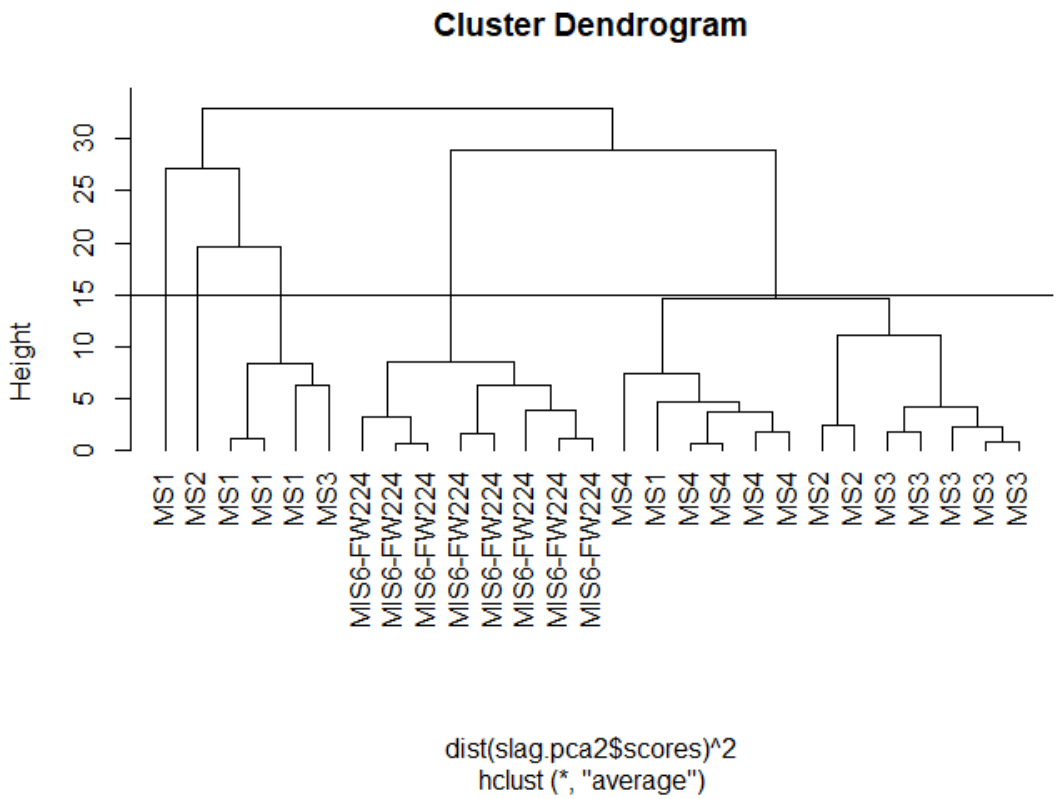
plot PCA scores and loadings

```
slag.pca2<-pcaplot(slag.wk3[els.slag2],1,2,slag.wk3$context,slag.wk3$context)
```



apply average-linkage cluster analysis

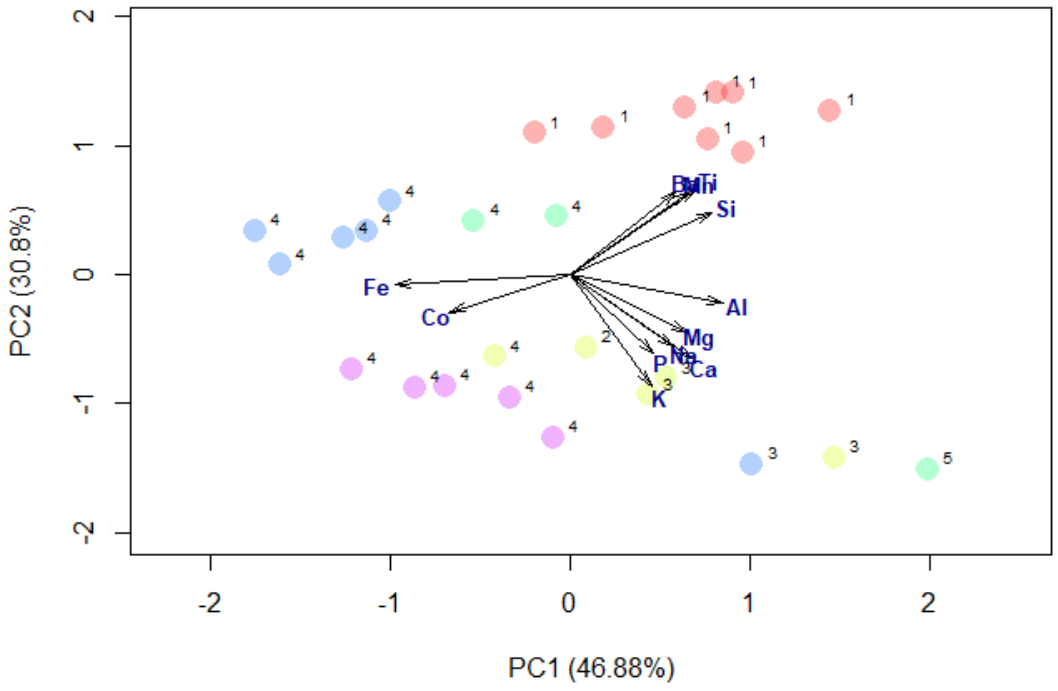
```
slag.hcl2<-hclust(dist(slag.pca2$scores)^2,"ave")
plot(slag.hcl2,hang=-1,labels=slag.wk3$context)
abline(h=15)
```




```
slag.clust2<-cutree(slag.hcl2,h=15)
```

plot PCA scores by cluster

```
pcaplot(slag.wk3[els.slag2],1,2,slag.wk3$context,slag.clust2)
```



rename clusters and define chemical groups

```
data.wk[names(slag.clust2[slag.clust2==5 |
                          slag.clust2==2]),"chem.group"]<-"XP slag outlier"
```

remove outliers from consideration

```
slag.wk4<-data.wk[names(slag.clust2[slag.clust2!=5 &
                                    slag.clust2!=2]),]

slag.wk4<-droplevels(slag.wk4)
```

select elements of interest for subset

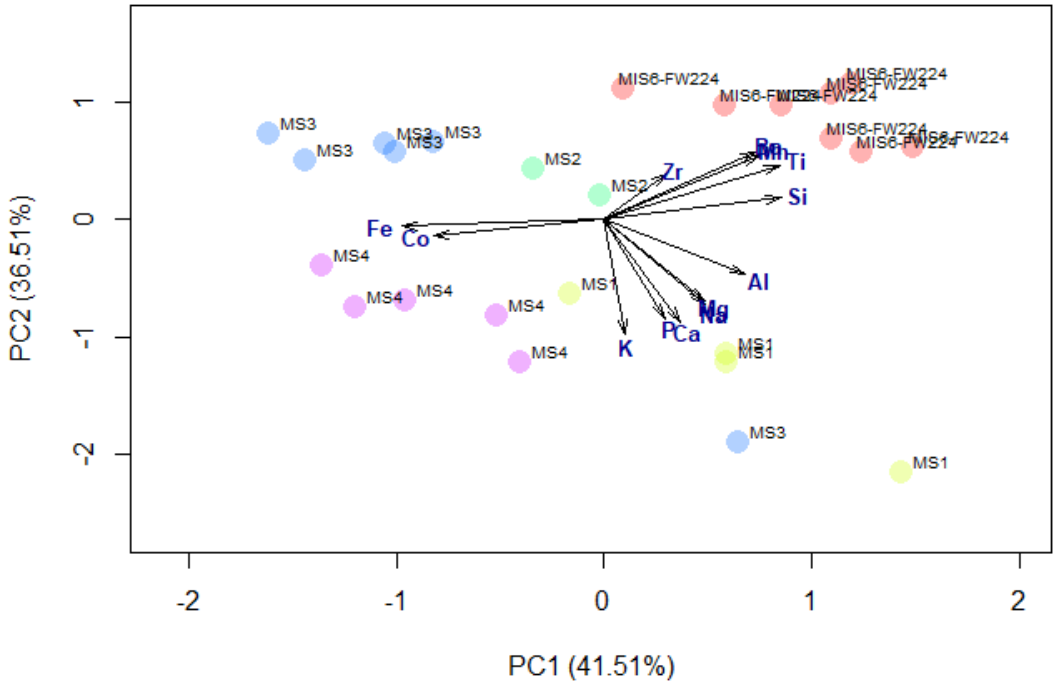
```
els.slag3<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(slag.wk4[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    els.slag3[n]<-els[t]
  }
}
```

transform data to subcompositions

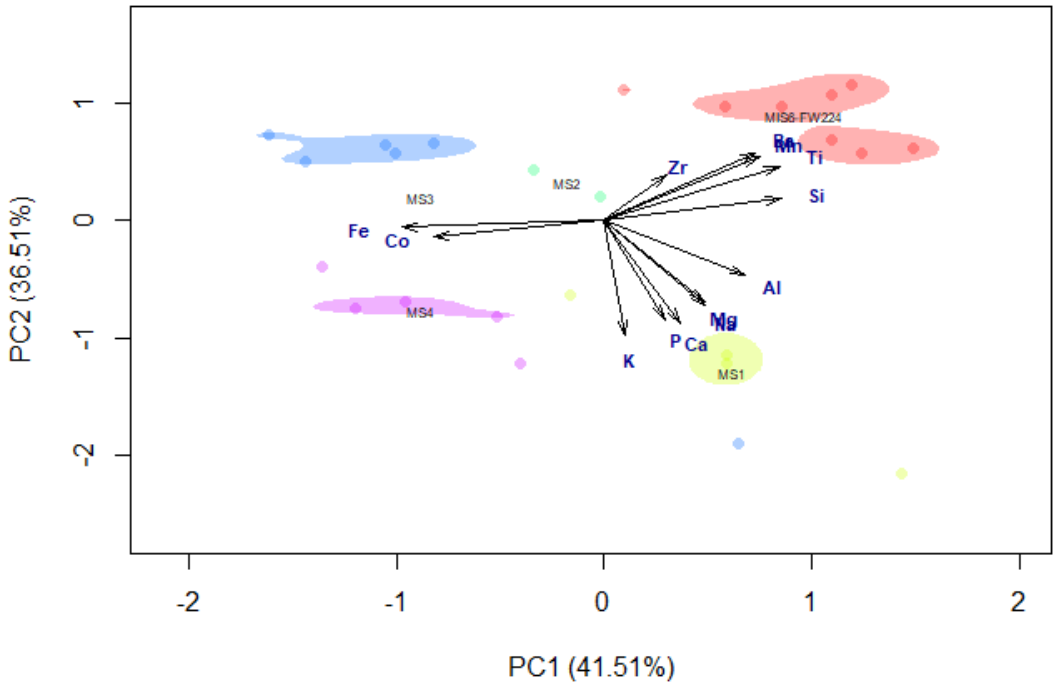
```
sum.slag.wk4<-apply(slag.wk4[els.slag3],1,sum)
slag.wk4[els.slag3]<-slag.wk4[els.slag3]/sum.slag.wk3*100
```

plot PCA scores and loadings

```
slag.pca3<-pcaplot(slag.wk4[els.slag3],1,2,slag.wk4$context,slag.wk4$context)
```

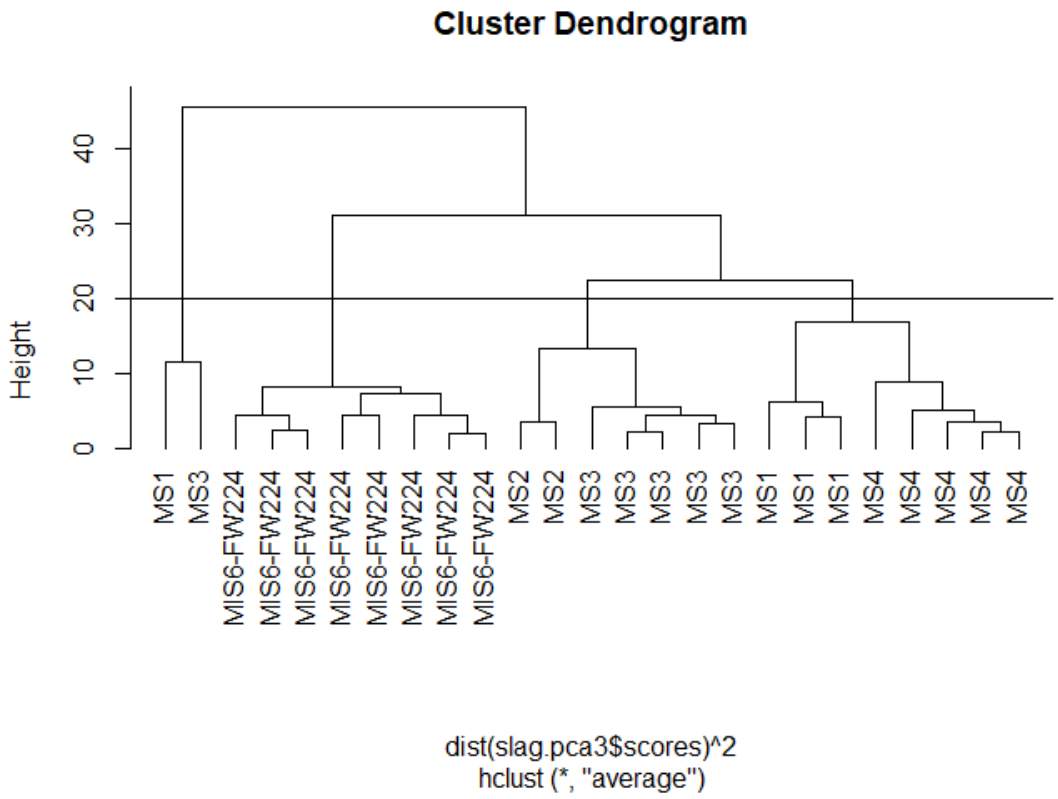



```
pcaplot.kde(slag.wk4[els.slag3],1,2,slag.wk4$context,slag.pca3$fc)
```



apply average-linkage cluster analysis

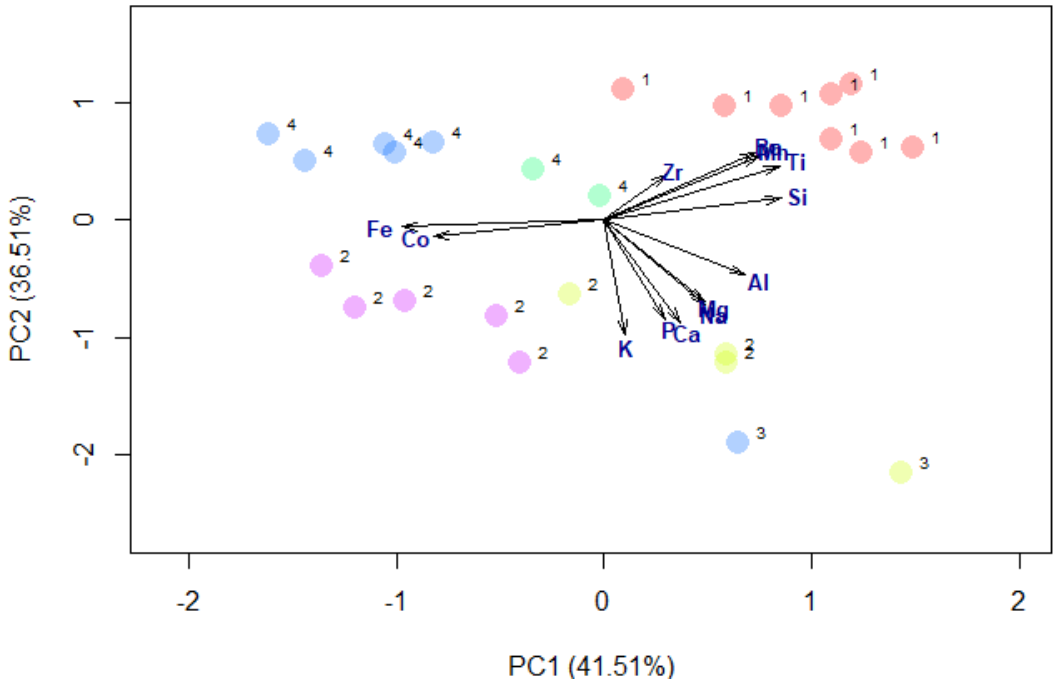
```
slag.hcl3<-hclust(dist(slag.pca3$scores)^2,"ave")
plot(slag.hcl3,hang=-1,labels=slag.wk4$context)
abline(h=20)
```



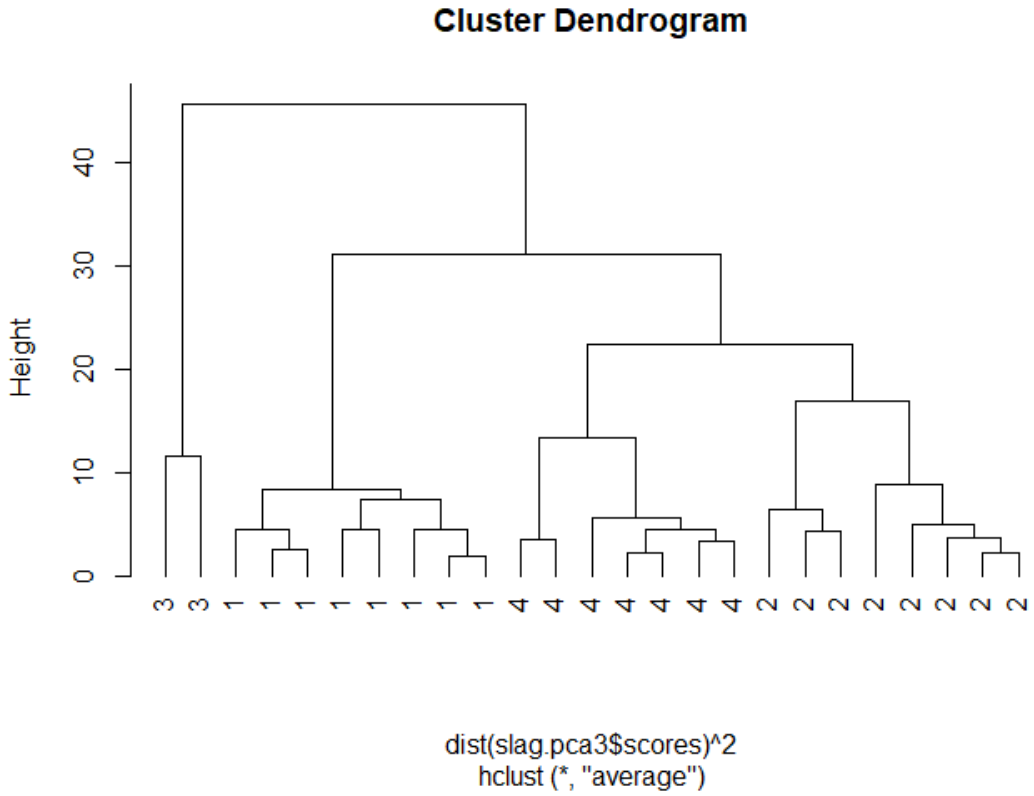

```
slag.clust3<-cutree(slag.hcl3,h=20)
```

plot PCA scores by cluster

```
pcaplot(slag.wk4[els.slag3],1,2,slag.wk4$context,slag.clust3)
```




```
plot(slag.hcl3,hang=-1,labels=slag.clust3)
```



rename clusters and define chemical groups

```
data.wk[names(slag.clust3[slag.clust3==3]),"chem.group"]<-"FA slag3"
data.wk[names(slag.clust3[slag.clust3!=3]),"chem.group"]<-"smelting slag"
```

reset slag data

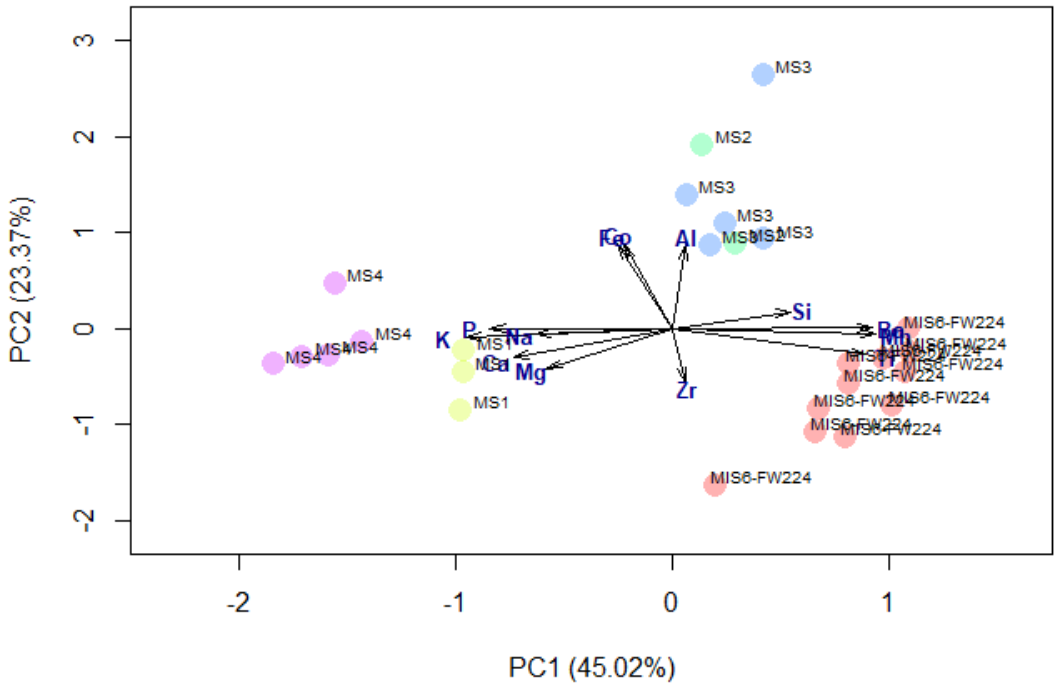
```
slag.wk4<-data.wk[data.wk$chem.group=="smelting slag" &
                    is.na(data.wk$chem.group)==F,]
slag.wk4<-droplevels(slag.wk4)
```

transform data to CLRs

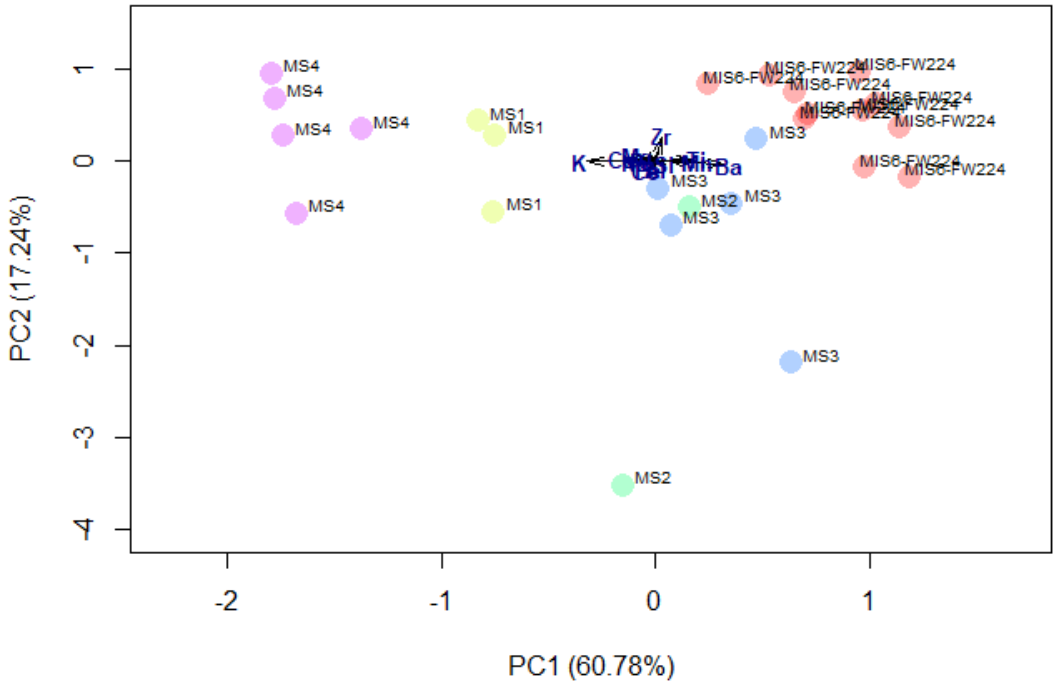
```
gm.slag1<-apply(slag.wk4[els.slag3],1,geomean)
clr.slag1<-log10(slag.wk4[els.slag3]/gm.slag1)
```

create standardized CLR and CLR-PCA plots

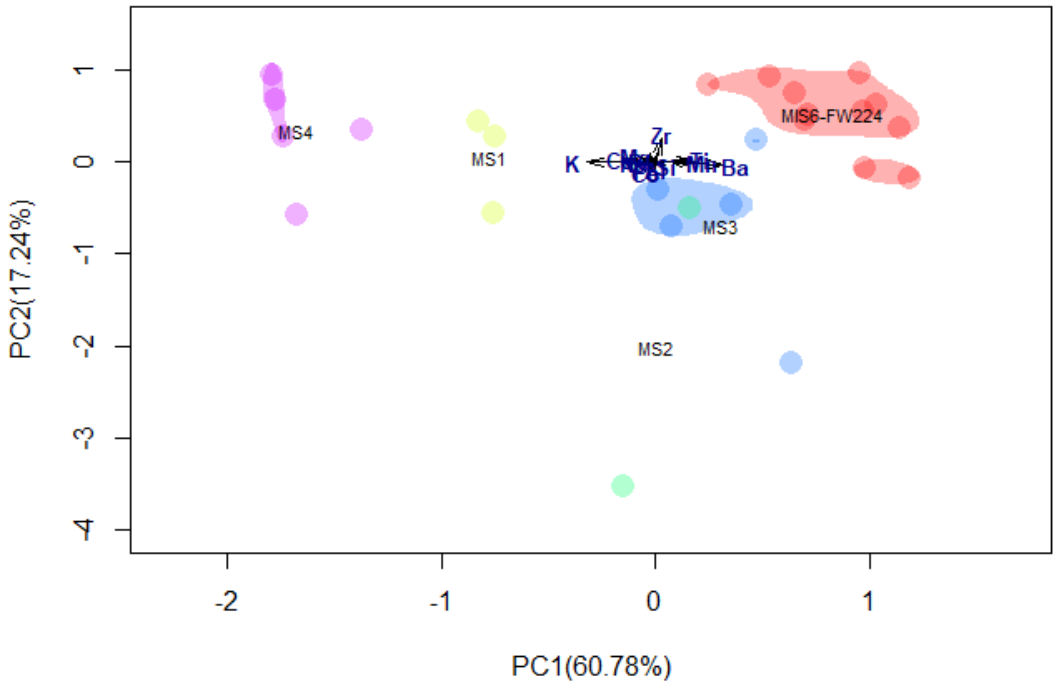
```
pcaplot(clr.slag1,1,2,slag.wk4$context,slag.wk4$context)
```



```
pcaplot.clr(clr.slag1,1,2,slag.wk4$context,slag.wk4$context)
```




```
pcaplot.clr.kde(clr.slag1,1,2,slag.wk4$context,slag.pca3$fc)
```



remove remaining outliers

```
slag.wk5<-data.wk[names(slag.clust3[slag.clust3!=3]),]
slag.wk5<-droplevels(slag.wk5)
```

select element of interest for subset

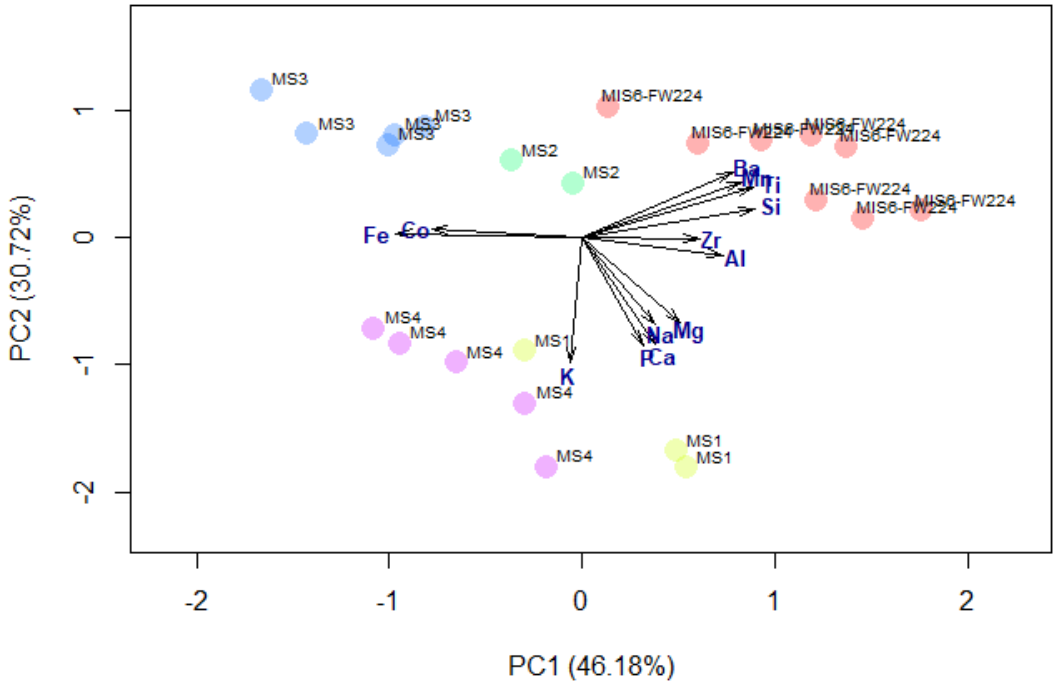
```
els.slag4<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(slag.wk5[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    els.slag4[n]<-els[t]
  }
}
```

transform data to subcompositions

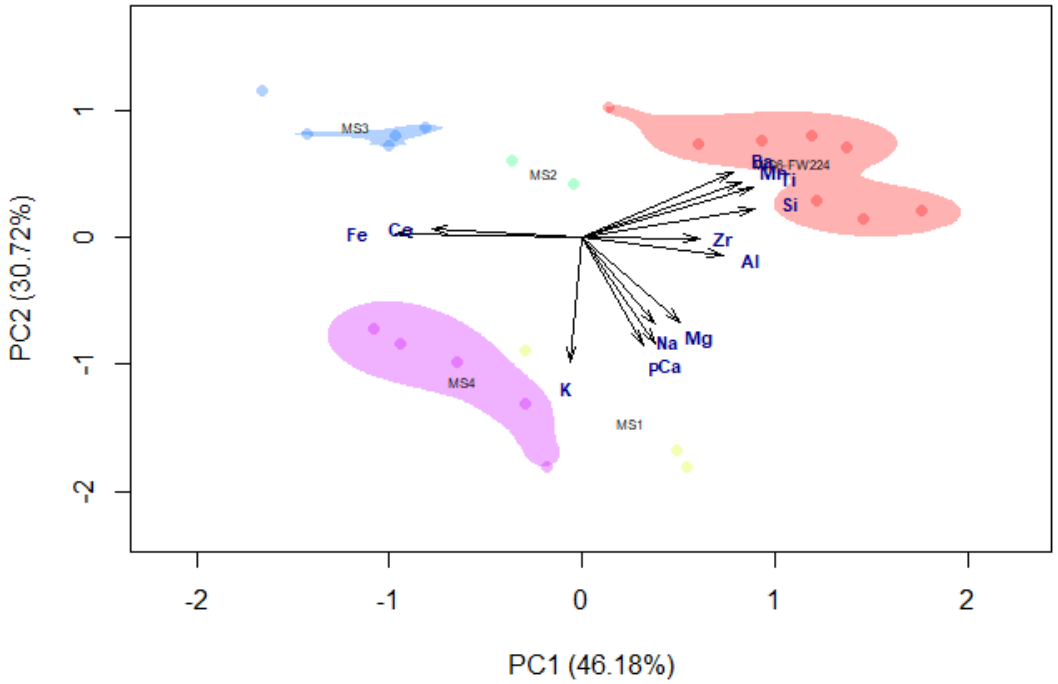
```
sum.slag.wk5<-apply(slag.wk5[els.slag4],1,sum)
slag.wk5[els.slag4]<-slag.wk5[els.slag4]/sum.slag.wk5*100
```

plot PCA scores and loadings

```
slag.pca4<-pcaplot(slag.wk5[els.slag4],1,2,slag.wk5$context,slag.wk5$context)
```




```
pcaplot.kde(slag.wk5[els.slag4],1,2,slag.wk5$context,slag.pca4$fc)
```

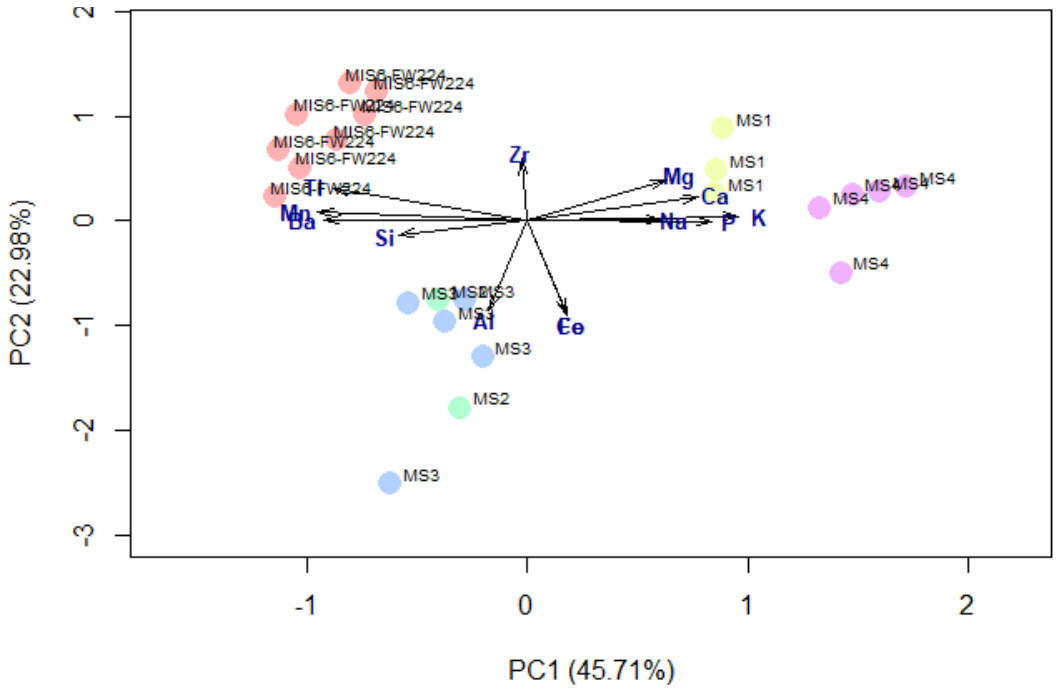


transform data to CLRs

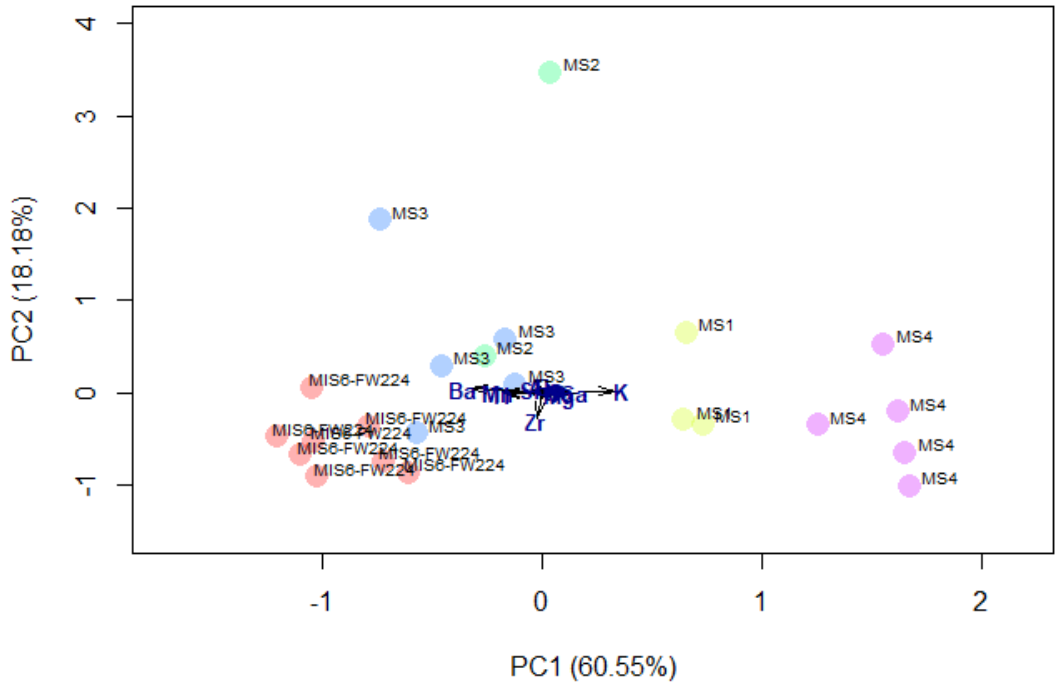
```
gm.slag2<-apply(slag.wk5[els.slag4],1,geomean)
clr.slag2<-log10(slag.wk5[els.slag4]/gm.slag2)
```

create standardized CLR and CLR-PCA plots

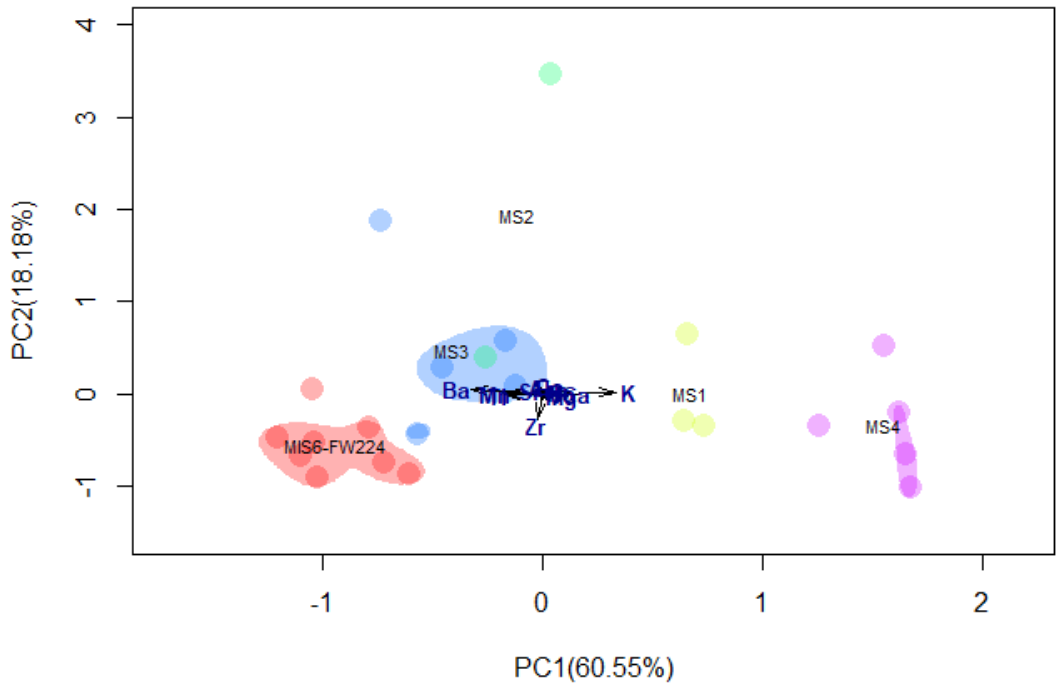
```
pcaplot(clr.slag2,1,2,slag.wk5$context,slag.wk5$context)
```



```
pcaplot.clr(clr.slag2,1,2,slag.wk5$context,slag.wk5$context)
```




```
pcaplot.clr.kde(clr.slag2,1,2,slag.wk5$context,slag.pca4$fc)
```



remove Fe and Si from elements of interest

```
els.slag5<-els.slag4[which(els.slag4!="Fe" &
                           els.slag4!="Si")]
```

select relevant data

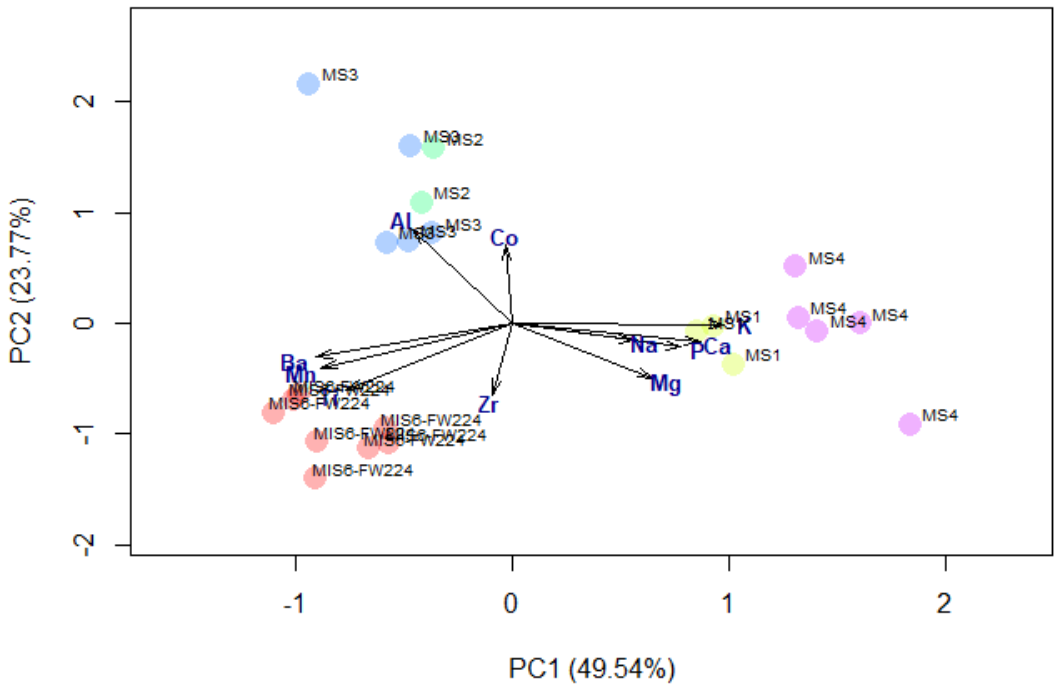
```
slag.wk5<-data.wk[names(slag.clust3[slag.clust3!=3]),]
slag.wk5<-droplevels(slag.wk5)
```

transform data to subcompositions

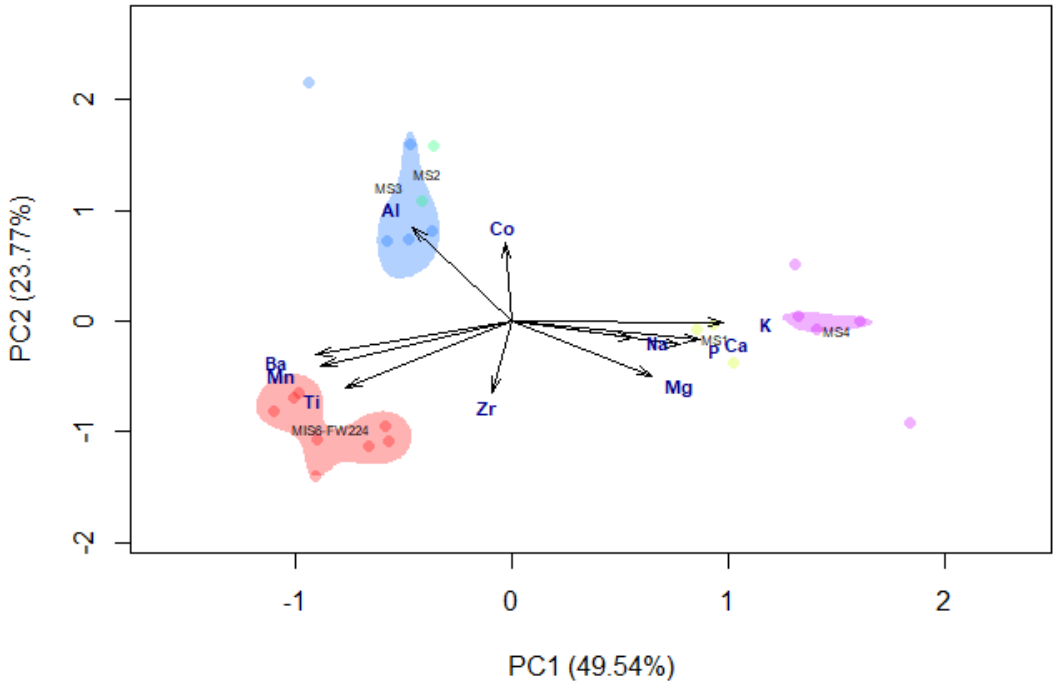
```
sum.slag.wk5<-apply(slag.wk5[els.slag5],1,sum)
slag.wk5[els.slag5]<-slag.wk5[els.slag5]/sum.slag.wk5*100
```

plot PCA scores and loadings

```
slag.pca5<-pcaplot(slag.wk5[els.slag5],1,2,slag.wk5$context,slag.wk5$context)
```



```
pcaplot.kde(slag.wk5[els.slag5],1,2,slag.wk5$context,slag.pca5$fc)
```



reset slag data

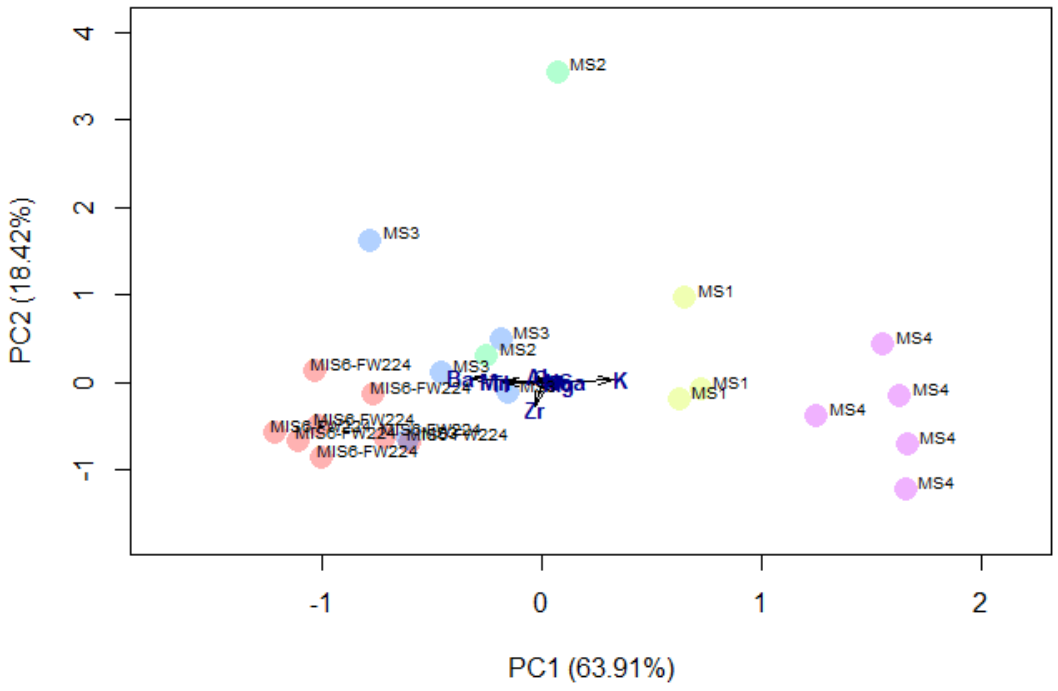
```
slag.wk5<-data.wk[names(slag.clust3[slag.clust3!=3]),]
slag.wk5<-droplevels(slag.wk5)
```

transform data to CLRs

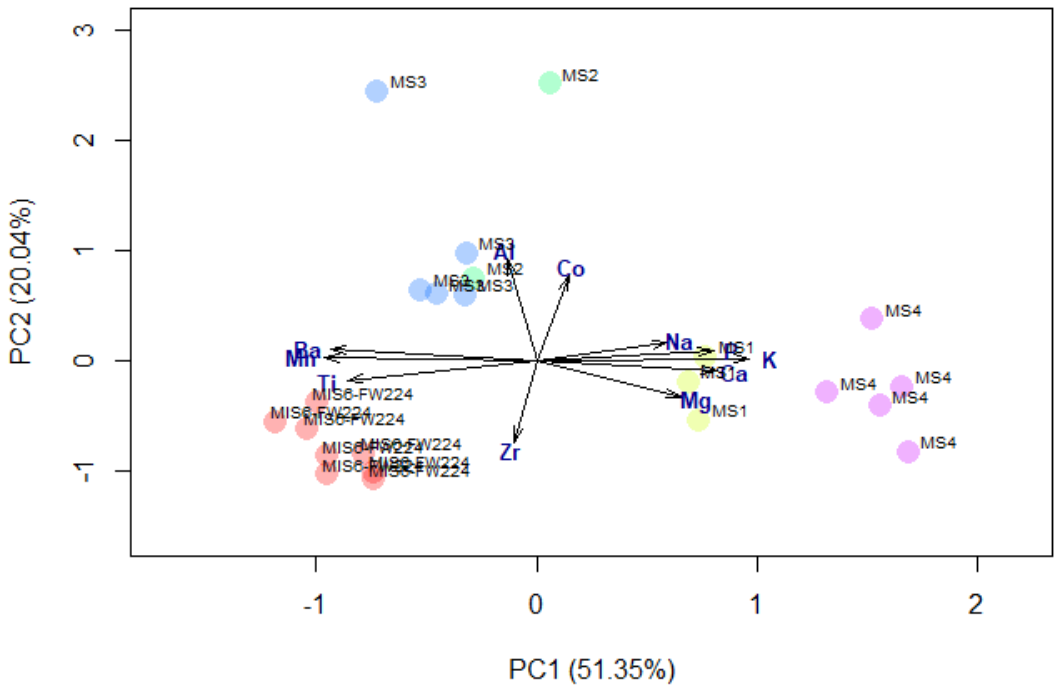
```
gm.slag3<-apply(slag.wk5[els.slag5],1,geomean)
clr.slag3<-log10(slag.wk5[els.slag5]/gm.slag3)
```

create standardized CLR and CLR-PCA plots

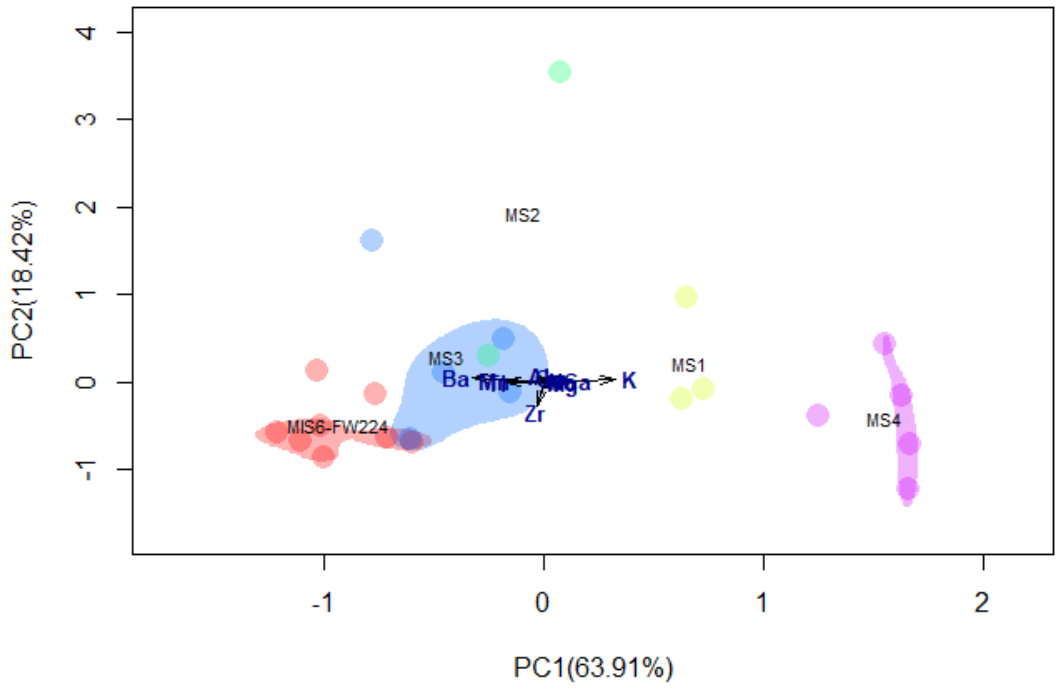
```
pcaplot.clr(clr.slag3,1,2,slag.wk5$context,slag.wk5$context)
```




```
pcaplot(clr.slag3,1,2,slag.wk5$context,slag.wk5$context)
```



```
pcaplot.clr.kde(clr.slag3,1,2,slag.wk5$context,slag.pca5$fc)
```



collect all slag data and create CSV file of the dataset

```
slag.all<-data.wk[data.wk$material=="slag" | data.wk$material=="slag.TC",]
write.csv(slag.all,file=file.choose())
```

Analysis of technical ceramic (TC) chemistry

select TC data

```
tc.wk<-data.wk[data.wk$material=="TC" & data.wk$context!="MS5",]
tc.wk<-droplevels(tc.wk)
```

select elements of interest

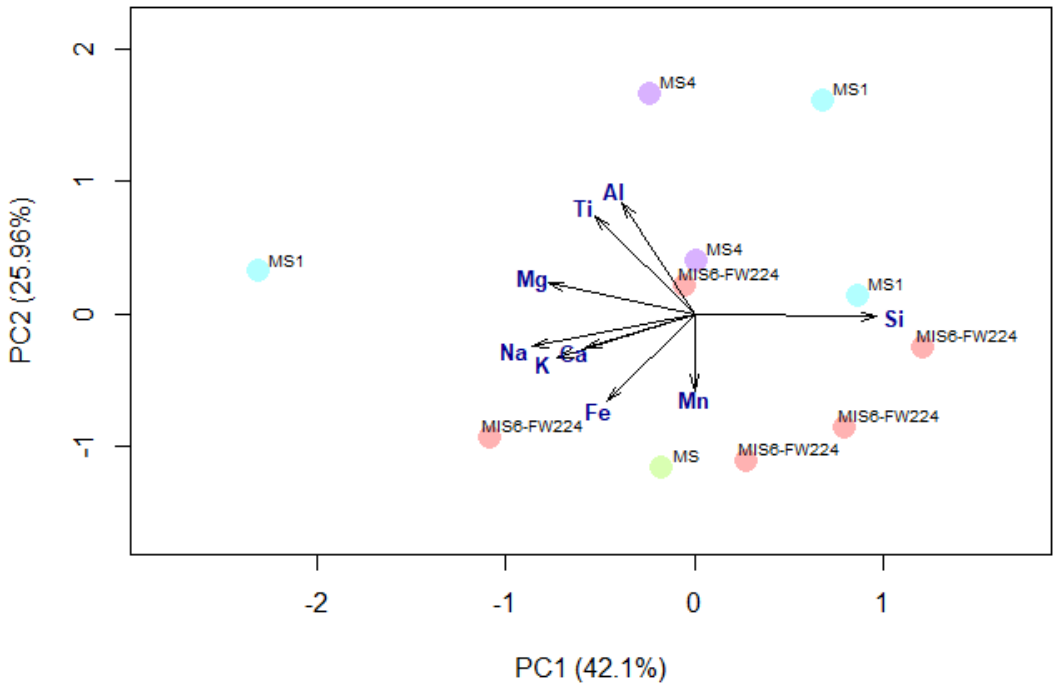
```
els.tc<-c()
n<-0
for(t in 1:length(els)){
  if(colMedians(as.matrix(tc.wk[els]))[t]<0.1){
    next
  }else{
    n<-n+1
    els.tc[n]<-els[t]
  }
}
```

transform data to subcompositions

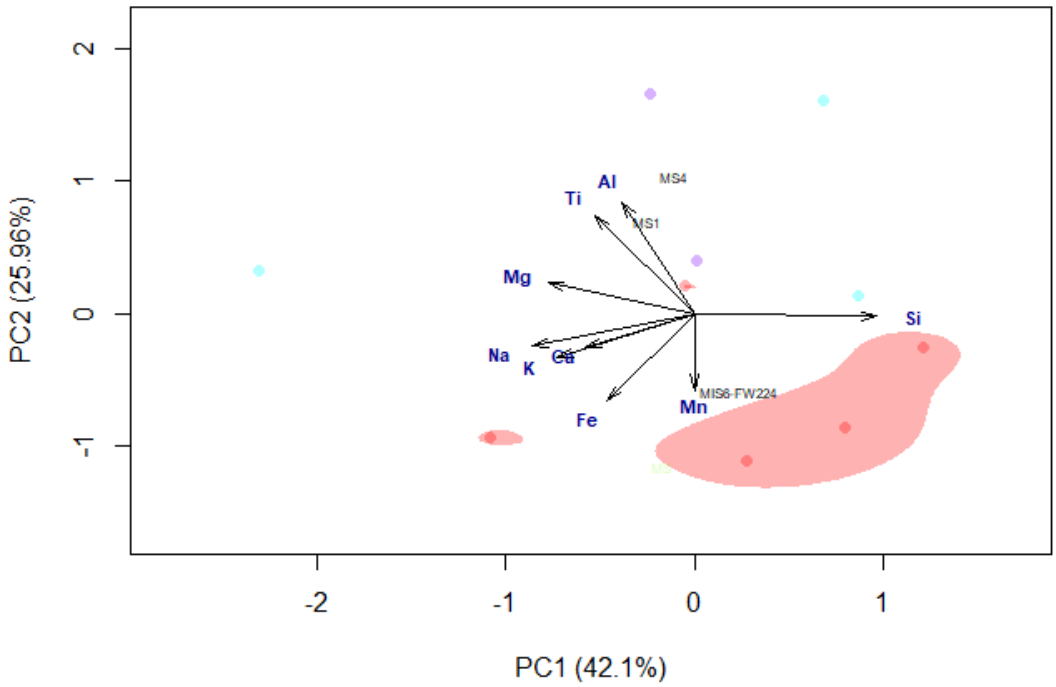
```
sum.tc.wk<-apply(tc.wk[els.tc],1,sum)
tc.wk[els.tc]<-tc.wk[els.tc]/sum.tc.wk*100
```

plot PCA scores and loadings

```
tc.pca<-pcaplot(tc.wk[els.tc],1,2,tc.wk$context,tc.wk$context)
```




```
pcaplot.kde(tc.wk[els.tc],1,2,tc.wk$context,tc.pca$fc)
```



reset TC data

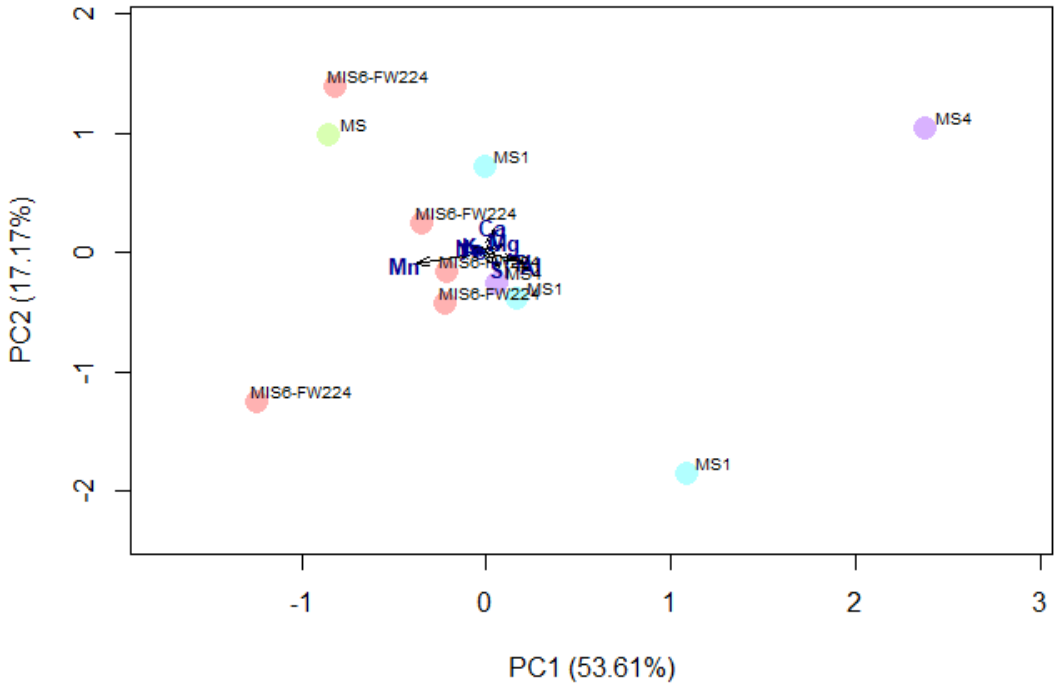
```
tc.wk<-data.wk[data.wk$material=="TC" & data.wk$context!="MS5",]
tc.wk<-droplevels(tc.wk)
```

transform data to CLRs

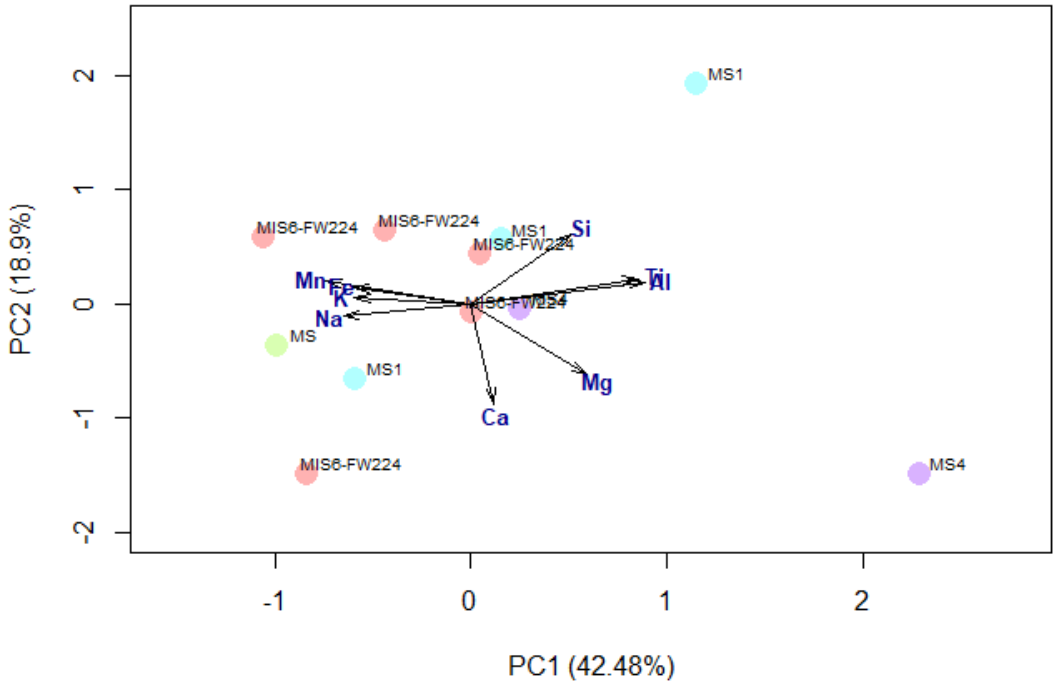
```
gm.tc<-apply(tc.wk[els.tc],1,geomean)
clr.tc<-log10(tc.wk[els.tc]/gm.tc)
```

plot standardized CLR and cLR-PCA plots

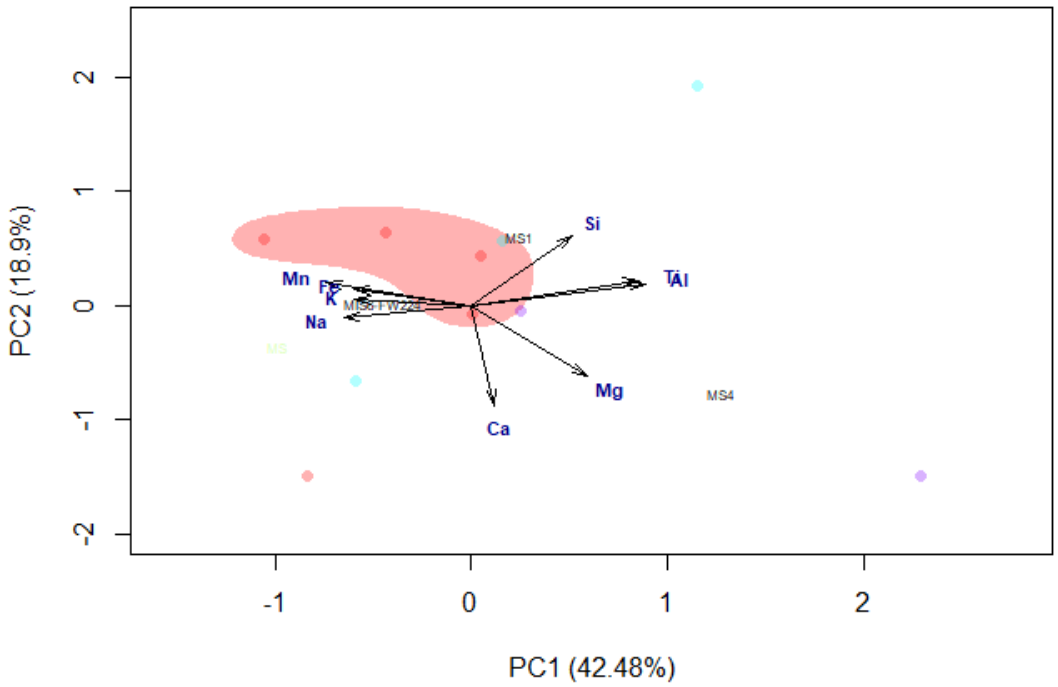
```
pcaplot.clr(clr.tc,1,2,tc.wk$context,tc.wk$context)
```



```
pcaplot(clr.tc,1,2,tc.wk$context,tc.wk$context)
```




```
pcaplot.kde(clr.tc,1,2,tc.wk$context,tc.pca$fc)
```



create CSV file of TC dataset

```
write.csv(tc.wk,file=file.choose())
```

Analysis of oxide data

Most traditional analysis of slag chemistry makes use of oxide rather than elemental wt % data. This approach assumes that most elements are bound with oxygen into common oxide molecules; an assumption validated by examination of specimen phase structure. These common oxides are calculated via stoichiometry using multipliers that define changes in mass between an element and its specified molecule.

The scripts below use stoichiometry to to convert elements to oxides. The multipliers used in this study can be found in the *.RData* file named ***SEM_element-oxide conversion factors***. Select oxides are then plotted into the FeO-SiO_2-Al_2O_3 ternary phase diagram. This diagram serves as a model for slag thermochemical behavior. Finally, the Reducible Iron Index (RII; Charlton et al., 2010) is calculated as an estimate of furnace redox conditions. The appropriate use of the RII depends on an accurate knowledge and consideration of ceramic and fuel-ash influences on slag chemistry. These were not available for this study, and the plots are provided solely for information and comparison.

convert elements to oxide wt %

```
ocf<-read.csv(file=file.choose(),header=T,row.names=1)
els.ocf<-colnames(ocf)
els.ocf<-els.ocf[which(els.ocf!="Hf" & els.ocf!="As")]
data.change<-data.wk[els.ox]

for(c in 1:ncol(data.change)){
  data.change[,c]<-as.numeric(data.change[,c])
}

data.wk[colnames(data.change)]<-data.change

ocf2<-t(ocf[els.ox])


ox.dat<-array(data=NA, dim=c(nrow(data.wk),ncol(data.wk[els.ox])))
for(r in 1:nrow(data.wk)){
  for(c in 1:ncol(data.wk[els.ox])){
    ox.dat[r,c]<-data.wk[els.ox][r,c]*ocf[els.ox][,c]
  }
}

values<-c(rep(NA,length(data.meta)), rep("wt %",length(ox.per)))
rownames(ox.dat)<-rownames(data.wk)
```

```
colnames(ox.dat)<-ox.per
ox.dat<-data.frame(ox.dat)
ox.dat.per<-round(ox.dat[ox.per],3)
ox.dat<-cbind(data.wk[data.meta], ox.dat.per)
colnames(ox.dat)<-c(data.meta,ox.per)
```

calculate RII

```
rii<-2.39*ox.dat$SiO2/(ox.dat$MnO+ox.dat$FeO)
ox.dat<-cbind(ox.dat,rii)
```

create CSV file for oxide dataset

```
write.csv(ox.dat,file=file.choose())
```

subset slag oxide data

```
slag.ox<-ox.dat[rownames(slag.wk5),]
slag.ox<-droplevels(slag.ox)
```

Plot slag data in FeO-SiO_2-Al_2O_3 ternary phase diagram

select and transform ternary data

```
tern.data<-slag.ox

tern.sum<-apply(tern.data[tern.els],1,sum)

tern.data["FeO.MnO"]<-tern.data["FeO"]+tern.data["MnO"]

tern.data["FeO.MnO"]<-(tern.data["FeO.MnO"])/tern.sum*100
tern.data["SiO2"]<-tern.data["SiO2"]/tern.sum*100
tern.data["Al2O3"]<-tern.data["Al2O3"]/tern.sum*100
```

define liquidus lines

Ternary liquidus lines display the shifting equilibrium freezing temperatures across phases compositions displayed in the ternary phase diagram. Here, those lines are smoothed across a series of points measured from the FeO-SiO_2-Al_2O_3 ternary system given in Levin et al. (1964). The points are given in the *.RData* file named ***FeO-SiO2-Al2O3 lines***.

```
tiso<-read.csv(file=file.choose(),header=T,row.names=1)
pl<-as.data.frame(tiso)

ternary(tern.data[tern.poles],tern.data$context,grid=T)
## $fc
## [1] "#FF00004D" "#CCFF004D" "#00FF664D" "#0066FF4D" "#CC00FF4D"
sq<-sqrt(3)/2

xpl1<-xspline(pl[1:9,"Al2O3"]+pl[1:9,"SiO2"]/2,
              pl[1:9,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
xpl2<-xspline(pl[10:27,"Al2O3"]+pl[10:27,"SiO2"]/2,
              pl[10:27,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
xpl3<-xspline(pl[28:71,"Al2O3"]+pl[28:71,"SiO2"]/2,
              pl[28:71,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
xpl4<-xspline(pl[72:76,"Al2O3"]+pl[72:76,"SiO2"]/2,
              pl[72:76,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
```

```
xpl5<-xspline(pl[77:78,"Al2O3"]+pl[77:78,"SiO2"]/2,
              pl[77:78,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
xpl6<-xspline(pl[79:83,"Al2O3"]+pl[79:83,"SiO2"]/2,
              pl[79:83,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
xpl7<-xspline(pl[84:92,"Al2O3"]+pl[84:92,"SiO2"]/2,
              pl[84:92,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
xpl8<-xspline(pl[93:99,"Al2O3"]+pl[93:99,"SiO2"]/2,
              pl[93:99,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
xpl9<-xspline(pl[100:107,"Al2O3"]+pl[100:107,"SiO2"]/2,
              pl[100:107,"SiO2"]*sq,
              1,
              lmitre=100,
              lend=1,
              ljoin=2,
              draw=F)
xpl10<-xspline(pl[108:113,"Al2O3"]+pl[108:113,"SiO2"]/2,
               pl[108:113,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
```

```
xpl11<-xspline(pl[114:117,"Al2O3"]+pl[114:117,"SiO2"]/2,
               pl[114:117,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl12<-xspline(pl[118:120,"Al2O3"]+pl[118:120,"SiO2"]/2,
               pl[118:120,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl13<-xspline(pl[121:123,"Al2O3"]+pl[121:123,"SiO2"]/2,
               pl[121:123,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl14<-xspline(pl[124:129,"Al2O3"]+pl[124:129,"SiO2"]/2,
               pl[124:129,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl15<-xspline(pl[130:133,"Al2O3"]+pl[130:133,"SiO2"]/2,
               pl[130:133,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl16<-xspline(pl[134:137,"Al2O3"]+pl[134:137,"SiO2"]/2,
               pl[134:137,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
```

```
xpl17<-xspline(pl[138:141,"Al2O3"]+pl[138:141,"SiO2"]/2,
               pl[138:141,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl18<-xspline(pl[142:145,"Al2O3"]+pl[142:145,"SiO2"]/2,
               pl[142:145,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl19<-xspline(pl[146:148,"Al2O3"]+pl[146:148,"SiO2"]/2,
               pl[146:148,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl20<-xspline(pl[149:151,"Al2O3"]+pl[149:151,"SiO2"]/2,
               pl[149:151,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl21<-xspline(pl[152:154,"Al2O3"]+pl[152:154,"SiO2"]/2,
               pl[152:154,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl22<-xspline(pl[155:157,"Al2O3"]+pl[155:157,"SiO2"]/2,
               pl[155:157,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
```

```
xpl23<-xspline(pl[158:160,"Al2O3"]+pl[158:160,"SiO2"]/2,
               pl[158:160,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl24<-xspline(pl[161:163,"Al2O3"]+pl[161:163,"SiO2"]/2,
               pl[161:163,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl25<-xspline(pl[164:169,"Al2O3"]+pl[164:169,"SiO2"]/2,
               pl[164:169,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
xpl26<-xspline(pl[170:171,"Al2O3"]+pl[170:171,"SiO2"]/2,
               pl[170:171,"SiO2"]*sq,
               1,
               lmitre=100,
               lend=1,
               ljoin=2,
               draw=F)
```

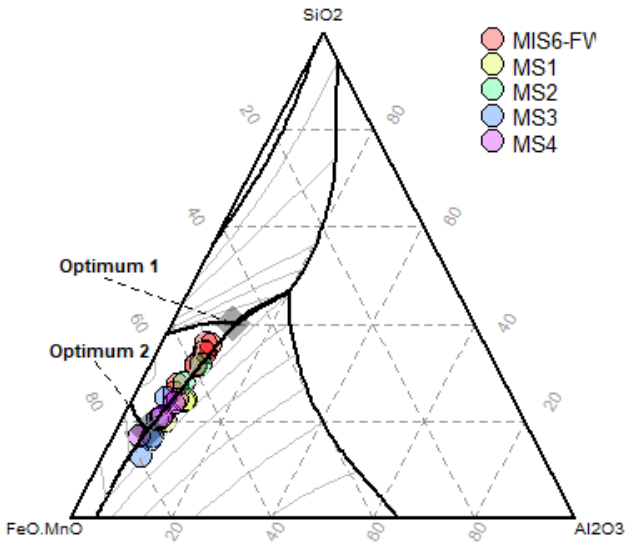
plot slag in the FeO-SiO_2-Al_2O_3 ternary diagram

Ternary eutectics are labeled optimum 1 and optimum 2 (Charlton et al., 2010), corresponding to theoretical optimal slag chemistries for iron production at opposite ends of the socioeconomic spectrum from low demand - low competition to high demand - high competition contexts.

```
tern.plot<-ternary(tern.data[tern.poles],tern.data$context,grid=T)

lines(xpl11$x,xpl11$y,lwd=1,col="gray")
lines(xpl12$x,xpl12$y,lwd=1,col="gray")
lines(xpl13$x,xpl13$y,lwd=1,col="gray")
lines(xpl14$x,xpl14$y,lwd=1,col="gray")
lines(xpl15$x,xpl15$y,lwd=1,col="gray")
```

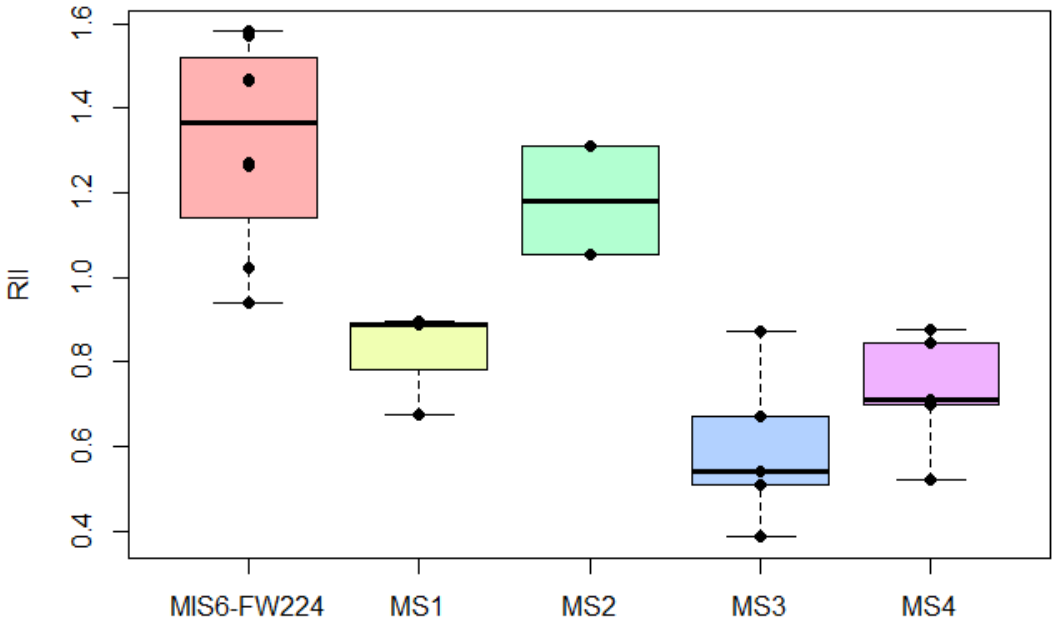
```
lines(xpl16$x,xpl16$y,lwd=1,col="gray")
lines(xpl17$x,xpl17$y,lwd=1,col="gray")
lines(xpl18$x,xpl18$y,lwd=1,col="gray")
lines(xpl19$x,xpl19$y,lwd=1,col="gray")
lines(xpl20$x,xpl20$y,lwd=1,col="gray")
lines(xpl21$x,xpl21$y,lwd=1,col="gray")
lines(xpl22$x,xpl22$y,lwd=1,col="gray")
lines(xpl23$x,xpl23$y,lwd=1,col="gray")
lines(xpl24$x,xpl24$y,lwd=1,col="gray")
lines(xpl25$x,xpl25$y,lwd=1,col="gray")
lines(xpl26$x,xpl26$y,lwd=1,col="gray")

lines(xpl1$x,xpl1$y,lwd=2,lmitre=100,lend=1,ljoin=2)
lines(xpl2$x,xpl2$y,lwd=2,lmitre=100,lend=1,ljoin=2)
lines(xpl3$x,xpl3$y,lwd=2,lmitre=100,lend=1,ljoin=2)
lines(xpl4$x,xpl4$y,lwd=2)
lines(xpl5$x,xpl5$y,lwd=2)
lines(xpl6$x,xpl6$y,lwd=2)
lines(xpl7$x,xpl7$y,lwd=2)
lines(xpl8$x,xpl8$y,lwd=2)
lines(xpl9$x,xpl9$y,lwd=2)
lines(xpl10$x,xpl10$y,lwd=2)

segments(0, 0, 1, 0,lwd=2)
segments(0, 0, 1/2, sqrt(3)/2,lwd=2)
segments(1/2, sq, 1, 0,lwd=2)

op1<-pl[76,]
op2<-pl[9,]
points(x=c(op2[,3]+op2[,4]/2),y=c(op2[,4]*sq),
       pch=23,
       col=NA,
       bg=rgb(0,0,0,max=255,alpha=90),
       cex=2.5
)

points(x=c(op1[,3]+op1[,4]/2),y=c(op1[,4]*sq),
       pch=23,
       col=NA,
       bg=rgb(0,0,0,max=255,alpha=90),
       cex=2.5
)
```

```
text(-0.02,0.45,"Optimum 1",adj=0,font=2,cex=0.7)
segments(0.07,0.43,op1[,3]+op1[,4]/2,op1[,4]*sq,lty=2)

text(-0.04,0.30,"Optimum 2",adj=0,font=2,cex=0.7)
segments(0.05,0.28,op2[,3]+op2[,4]/2,op2[,4]*sq,lty=2)


legend(0.8,0.9,
       levels(slag.ox$context),
       col="black",
       pt.bg = tern.plot$fc,
       pch=21,
       bty="n",
       cex=0.8,
       pt.cex=2)
```



Plot RII boxplot for each experimental and archaeological context

```
boxplot(slag.ox$rii~slag.ox$context, ylab="RII",col=tern.plot$fc)
points(slag.ox$rii~slag.ox$context,
       pch=21,
       cex=1,
       bg="black")
```



Acknowledgements

Permission for the research to be carried out at Meroe is granted by the National Corporation for Antiquities and Museums in Sudan, and for their assistance and continual support we are sincerely thankful. A number of organisations provide collaborations and funding through with various aspects of this research have been greatly assisted. These include the Qatar-Sudan Archaeology Project, UCL Qatar, the University of Khartoum, and the British Institute in Eastern Africa. Michael Charlton's participation and research was supported by a Marie Curie International Incoming Fellowship (PIIF-GA-2013-624448) within the 7th European Community Framework Programme as part of the IRONWORKS project. All samples were prepared by Loic Boscher. Additional support with sample selection and preparation was provided by Philip Connelly, Tom Birch, and Fareed Alshishani. Special thanks go to smelters Jake Keen and Lee Sauder for conducting the experiments and Suleiman Awad Suleiman, Thomas Scheibner and Saskia Büchner for constant and multifaceted support throughout the project.

References:

Charlton, M.F., Crew, P., Rehren, T., Shennan, S.J., 2010. Explaining the evolution of ironmaking recipes – An example from northwest Wales. J. Anthropol. Archaeol. 29, 352–367. doi:10.1016/j.jaa.2010.05.001

Humphris, J., Rehren, T., 2014. Iron production and the Kingdom of Kush: an introduction to UCL Qatar's research in Sudan, in: Lohwasser, A., Wolf, P. (Eds.), Ein Forscheleben Zwischen Den Welten. Sonderheft, Berlin, pp. 177–190.

Levin, E.M., Robbins, C.R., McMurdie, H.F., 1964. Phase Diagrams for Ceramists. The American Ceramic Society.

Plumlee, G., 1998a. Basalt, Columbia River, BCR-2.

Plumlee, G., 1998b. Basalt, Hawaiian Volcanic Observatory, BHVO-2.

R Core Team, 2017. R: a language and environment for statistical computing.

Schindelin, J., Rueden, C.T., Hiner, M.C., Eliceiri, K.W., 2015. The ImageJ ecosystem: An open platform for biomedical image analysis. Mol. Reprod. Dev. 82, 518–529. doi:10.1002/mrd.22489

Smith, D.B., 1998. Icelandic Basalt, BIR-1.

Templ, M., Hron, K., Filzmoser, P., 2011. robCompositions: an R-Package for robust statistical analysis of compositional data, in: Pawlowsky-Glahn, V., Buccianti, A. (Eds.), Compositional Data Analysis. John Wiley & Sons, Chichester, pp. 341–355.

Venables, W.N., Ripley, B.D., 2002. Modern Applied Statistics with S, Fourth. ed. Springer, New York.